



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA POLITÉCNICA SUPERIOR**

Departamento de Sistemas Informáticos

INGENIERÍA EN INFORMÁTICA

PROYECTO FINAL DE CARRERA

coCTT: Modelado de tareas en un entorno colaborativo

Blanca Azahara Arribas Simarro

Septiembre, 2007



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA POLITÉCNICA SUPERIOR**

Departamento de Sistemas Informáticos

INGENIERÍA EN INFORMÁTICA

PROYECTO FINAL DE CARRERA

Presentado por: Blanca Azahara Arribas Simarro

Dirigido por: Francisco Montero Simarro

Septiembre, 2007

Agradecimientos

Quisiera agradecer en primer lugar a Francisco el apoyo prestado en la elaboración de este proyecto, ya que sin él, el desarrollo de este trabajo hubiese sido mucho más costoso.

También debo mostrar mi agradecimiento a todos mis profesores por las enseñanzas recibidas durante mi periodo universitario.

Me gustaría agradecer también a todos mis amigos y compañeros de clase por el apoyo prestado en cada momento.

Y por último me gustaría agradecer sinceramente a mis padres, a mi hermana y a Víctor su comprensión, ayuda y ánimo durante todo este tiempo.

Resumen

Para llevar a cabo cualquier proyecto de Ingeniería deben intervenir un grupo heterogéneo de personas y a la cantidad de ventajas que este hecho acarrea, se le suman algunas desventajas que son subsanadas dividiendo las tareas entre los miembros del grupo, concertando reuniones periódicas...etc. En la construcción de interfaces de usuario y más concretamente en el análisis de tareas ocurre algo parecido. El análisis de tareas es un término que cubre diferentes técnicas orientadas a describir las interacciones entre las personas y los entornos de una manera sistemática definiéndose como *el estudio de lo que un usuario tiene que realizar en términos de acciones y/o procesos cognitivos para conseguir un objetivo*. Es por tanto una metodología que está soportada por un conjunto de técnicas para ayudar en el proceso analítico de recogida de información, organizarlo y usarlo para realizar valoraciones o decisiones de diseño. Este proceso es complejo y requiere de numerosas personas para llevarlo a cabo.

Las herramientas conocidas hasta la fecha para llevar a cabo estas labores son monousuario, siendo necesaria la integración de las tareas llevadas a cabo por cada usuario en un diseño final.

La finalidad de este proyecto es la realización de una herramienta que facilite la comunicación e intercambio de información a través de Internet en modo síncrono entre miembros de un grupo para la realización de modelado de tareas, debiendo facilitar el seguimiento del progreso del grupo y la integración final del trabajo.

Albacete, septiembre de 2007

Índice de contenidos

Agradecimientos	I
Resumen	III
Índice de contenidos	V
Índice de tablas	VII
Índice de figuras	IX
Capítulo 1	
Introducción.....	1
1.1. Motivación.....	1
1.2. Problema.....	2
1.3. Hipótesis de trabajo	3
1.4. Objetivos.....	3
1.5. Estructura de la memoria.....	4
Capítulo 2	
Estado del arte	7
2.1. Introducción.....	7
2.2. Dinámicas del trabajo en grupo	8
2.3. Conceptos CSCW y Groupware	19
2.3.1. Trabajo cooperativo soportado por computador.....	19
2.3.2. Groupware	21
2.4. Modelado de tareas. Notación CTT.....	28
2.4.1. Ejemplo CTT	36
2.5. Análisis y conclusiones	38
Capítulo 3	
coCTT: Un entorno colaborativo para el modelado de tareas basado en CTT e IdealXML	41
3.1. Herramientas disponibles en el modelado de tareas.....	41
3.2. El editor ConcurTaskTrees: Ctte	41
3.3. IdealXML	42
3.4. coCTT.....	43
Capítulo 4	
Detalles de implementación.....	45
4.1. Introducción.....	45
4.2. Situación actual	46
4.3. alcance y justificación	47

4.4. objetivos	48
4.5. Especificación de requisitos software	49
4.6. Vista de casos de uso	53
4.6.1. Actores.....	53
4.6.2. Diagramas de casos de uso	53
4.7. Matriz de rastreabilidad objetivos / requisitos.....	57
4.8. Diagramas de Secuencia	58
4.8.1. Inicio de sesión de un usuario en el sistema.....	58
4.8.2. Comunicar	60
4.8.3. Incorporación tardía.....	60
4.8.4. Finalizar sesión	61
4.9. Diseño del sistema.....	62
Capítulo 5	
Caso de estudio.....	65
5.1. CoCTT: Ejemplo 1. Ejemplo práctico	65
5.2. CoCTT: Ejemplo 2. Propuesta de resolución ante un problema ambiguo	73
5.3. CoCTT: Ejemplo 3. Aprendizaje entre usuarios	75
Capítulo 6	
Trabajo futuro y conclusiones	79
6.1. Conclusiones.....	79
6.2. Trabajo Futuro	80
Bibliografía.....	85
Apéndice A	
Manual de usuario de coCTT	89

Índice de tablas

Tabla 2-1 Taxonomía espacio-temporal del groupware	24
Tabla 2-2. Clasificación del groupware [Johansen, 1988; Ortega, 2001]	24
Tabla 2-3. Taxonomía de Dyson, según el objetivo principal del Groupware	26
Tabla 2-4. Tipos de tareas definidas en la notación CTT	34
Tabla 2-5. Operadores temporales definidos en la notación CTT	36
Tabla 4-1. Datos específicos en la entrada al sistema	49
Tabla 4-2. Datos específicos en la comunicación entre usuarios	50
Tabla 4-3. Iconos asociados a las tareas y operadores temporales	51
Tabla 4-4. Acciones y datos asociados al modelado de tareas	51
Tabla 4-5. Datos específicos en la incorporación de tareas tardías	52
Tabla 4-6. Datos específicos en la gestión de Documentos	52
Tabla 4-7. Datos específicos a la salida del sistema.....	52
Tabla 4-8. Caso de uso UC-0001: iniciar	54
Tabla 4-9. Caso de uso UC-0002: comunicar	55
Tabla 4-10. Caso de uso UC-0003: elaborar modelo	55
Tabla 4-11. Caso de uso UC-0004: incorporación tardía	55
Tabla 4-12. Caso de uso UC-0005: Gestión del modelo	56
Tabla 4-13. Caso de uso UC-0006: Personalizar char.....	56
Tabla 4-14. Caso de uso UC-0007: Finalizar	56
Tabla 4-15. Matriz de rastreabilidad. Objetivos y requisitos de información	57
Tabla 4-16. Matriz de rastreabilidad: Objetivos y requisitos funcionales.....	58

Índice de figuras

Figura 2-1. Posible ciclo de desarrollo de una aplicación tradicional.....	8
Figura 2-2. Ciclo de desarrollo de una aplicación multiusuario.....	9
Figura 2-3. Principales funciones del trabajo colaborativo (DAV, 1998).....	10
Figura 2-4. Elementos del proceso de comunicación.....	10
Figura 2-5. Aspectos más relevantes de un grupo.....	12
Figura 2-6. Dinámica de grupo: Brainstorming.....	13
Figura 2-7. Dinámica de grupo: círculos de calidad.....	14
Figura 2-8. Juego colaborativo Second Life.....	15
Figura 2-9. Dinámica de grupo: foro.....	15
Figura 2-10. Dinámica de grupo: entrevista.....	16
Figura 2-11. Dinámica de grupo: mesa redonda.....	17
Figura 2-12. Dinámica de grupo: discusión dirigida.....	17
Figura 2-13. Dinámica de grupo: philips 6/6.....	18
Figura 2-14. Evolución de los sistemas software [Grudin, 1994].....	20
Figura 2-15. Clasificación de herramientas colaborativas en función del espacio/tiempo.....	24
Figura 2-16 Modelado de tareas correspondiente al acceso a la información de un museo.....	37
Figura 2-17. Operadores temporales: significado en el diálogo secuencial.....	37
Figura 2-18. Operadores temporales: significado en el diálogo elección.....	38
Figura 2-19. Operadores temporales: significado en el diálogo concurrencia.....	38
Figura 2-20. Operadores temporales: significado en el diálogo desactivar.....	38
Figura 3-1. Editor CTTE.....	42
Figura 3-2. Editor de modelo de tareas incluido en IDEALXML.....	43
Figura 4-1. Desarrollo de software siguiendo el Proceso Unificado de Rational (Jacobson et al., 1999; Krutchen et al., 2000).....	45
Figura 4-2. Icono asociado a un actor/usuario.....	53
Figura 4-3. Diagrama de casos de uso del subsistema cliente.....	54
Figura 4-4. Diagrama de casos de uso del subsistema servidor.....	57
Figura 4-5. Diagrama de secuencia correspondiente al inicio de sesión.....	59
Figura 4-6. Diagrama de secuencia asociado al proceso de comunicación.....	60
Figura 4-7. Diagrama de secuencia correspondiente a un escenario de incorporación tardía.....	61
Figura 4-8. Diagrama de secuencia correspondiente al cierre de sesión.....	61
Figura 4-9. Diagrama de paquetes del subsistema cliente.....	62
Figura 4-10. Diagrama de paquetes del subsistema servidor.....	63
Figura 5-1. Planteamiento del problema mediante la aplicación CoCTT.....	66
Figura 5-2. Resolución del problema con la herramienta CoCTT (Ejemplo 1).....	70
Figura 5-3. Resolución del problema con la herramienta CoCTT (Ejemplo 2).....	70
Figura 5-4. Resolución del problema con la herramienta coCTT (Ejemplo 3).....	71
Figura 5-5. Resolución del problema con la herramienta coCTT (Ejemplo 4).....	71
Figura 5-6. Consulta de la especificación XML asociada a un modelo de tareas.....	72
Figura 5-7. Invocación a la operación Guardar.....	73
Figura 5-8. Situación de posible ambigüedad en la especificación de un modelo de tareas.....	73
Figura 5-9. Caso práctico, análisis de tareas: Exposición de problema de ambigüedad.....	74

Figura 5-10. Caso práctico, análisis de tareas: resolución gráfica de problemas de ambigüedad.....	75
Figura 5-11. Caso práctico de análisis de tareas: exposición de problemas (1)	76
Figura 5-12. Caso práctico de análisis de tareas: resolución de problemas (2).....	76
Figura 5-13. Caso práctico de análisis de tareas: resolución de problemas (3).....	77
Figura 5-14. Caso práctico de análisis de tareas: resolución de problemas (4).....	77
Figura 5-15. Caso práctico, análisis de tareas: resolución gráfica de problemas (5)	78

Capítulo 1

Introducción

“El individuo adquiere saber en el proceso de la obtención de la experiencia mediante la propia acción y mediante su experiencia de las relaciones sociales”.

Franz Droge

1.1. MOTIVACIÓN

Está comprobado que el trabajo en grupo permite adoptar decisiones y llegar a conclusiones más ricas que aquellas que son producto del pensamiento de un solo individuo, además puesto que “cuatro ojos ven más que dos”, esta forma de trabajo aumenta la calidad de las decisiones que son tomadas por un entorno colectivo disminuyendo la incertidumbre y el riesgo a equivocarse en la toma de decisiones.

Al igual que los animales cuando están juntos trabajan más; las hormigas por ejemplo, construyen sus nidos a una velocidad 3 veces mayor cuando trabajan en grupos, los humanos y en especial en el ámbito de la informática ocurre lo mismo. Por ello, a la hora de diseñar software tenemos que tener en cuenta tanto los aspectos funcionales que ya nos aportan las herramientas tecnológicas actuales como los aspectos no funcionales, que pueden proporcionar un valor añadido de calidad.

Ha sido la aparición de Internet la que ha potenciado la relación entre las personas que usan el ordenador como medio de trabajo apartando a un lado la idea del ordenador como un único instrumento individual en el que el trabajo es llevado a cabo por una única persona. Esto nos permite abordar aspectos menos reconocidos pero también importantes a la hora de construir software de calidad, como son el modo en el que colabora un usuario con otros usuarios del sistema, como se comunican entre ellos, qué tipo de coordinación se sigue, aspectos espacio-temporales... etc.

Para llevar a cabo un proyecto software real, es rara la vez que se trabaja de forma individual y si se hace es en proyectos pequeños de menor complejidad, para los demás, la forma de trabajo es en común, siendo especialmente proclives a ser abordadas en equipo las tareas relacionadas con actividades de diseño y análisis. Aprender de los demás, compartir ideas, tomar decisiones, planear el trabajo, simultanear tareas... etc. son labores del día a día de una organización por lo que se debe de construir software que se adapte a estas necesidades.

1.2. PROBLEMA

Tanto en la construcción de interfaces de usuario como en las diferentes etapas que se siguen en el desarrollo de una aplicación software(requisitos, análisis, diseño, implementación, pruebas) intervienen diferentes personas que deben comunicarse entre sí. El problema aparece cuando el equipo de desarrollo que está trabajando sobre un mismo proyecto se encuentra en lugares diferentes. Como es ha comentado anteriormente, este es un hecho a tener en cuenta, pues para que un sistema informático resulte útil y tenga éxito, deben considerarse diferentes factores; las personas que trabajan en él, su localización, la tecnológica, la organización de trabajo y cultura organizacional y que todo esto se encuentre coordinado.

Debido a que algunos conceptos como CSCW o Groupware son relativamente recientes no existen en la actualidad muchas aplicaciones que implementen la filosofía de trabajo en grupo. La falta de estándares ha causado que este sea un proceso lento aunque cada vez son más las aplicaciones colaborativas que están surgiendo y se cree que la tendencia es que este número siga aumentando.

Cada vez se está investigando y aportando más ideas sobre este hecho que antes se intentaba solucionar con técnicas costosas tanto económica como temporalmente: preparación de reuniones que implican la movilidad del personal hasta un lugar común, conversaciones telefónicas, listas de correo... etc.

De forma más concreta, para abordar el proceso de análisis y especificación de una aplicación software se conocen diferentes entornos como CTTE (Paternó 1999) (ConcurTaskTrees Environment) o idealXML (Montero 2005) de las que se hablará en capítulos siguientes, pero estos son aplicaciones de escritorio pensada para un

único usuario. En el modelado de tareas por lo general colaboran un grupo de personas y estas aplicaciones se pueden ver limitadas ante este hecho.

1.3. HIPÓTESIS DE TRABAJO

En este Proyecto Final de Carrera se propone la realización de una herramienta que facilite la especificación de modelos de tareas en un entorno colaborativo. Con ella se pretende solucionar los problemas que implica el trabajo en equipo cuando los miembros que lo forman no se encuentran en un mismo lugar y llenar el vacío existente en torno a este tipo de aplicaciones cuando se trata de realizar modelos de tareas.

La herramienta construida es específica para la notación CTT (se profundizará sobre esta notación en capítulos posteriores), e intenta abordar además de las características de calidad que todo sistema software requiere, otros aspectos de vital importancia que intentarán proporcionar una comunicación fluida entre usuarios a la hora de llevar a cabo tareas de modelado.

1.4. OBJETIVOS

El principal objetivo que persigue la realización de este Proyecto es aportar valores añadidos de calidad a la hora de realizar modelos de tareas en la construcción de interfaces de usuario. Para conseguir esta finalidad, se han marcado una serie de metas, las cuales pretenden conseguir una solución satisfactoria. De forma más detallada se pretende:

- Dar una idea general del modo en el que trabajan las organizaciones en la actualidad destacando la tendencia del trabajo en equipo o grupo.
- Identificar los problemas que acarrearán la realización del trabajo de modo grupal.
- Eliminar el problema de la distancia a la hora de formar grupos de trabajo en organizaciones que no tienen por qué estar solo en una misma localización.
- Afianzar conceptos como CSCW o Groupware.

- Determinar las carencias existentes al día de hoy, la falta de herramientas y métodos que soporten la elaboración de herramientas colaborativas.
- Destacar la importancia del modelado de tareas en la construcción de interfaces de usuario.
- Hacer un recorrido por las técnicas de modelado de tareas actuales haciendo un estudio mas profundo sobre la notación CTT destacando por qué ha sido la notación utilizada para realizar el Proyecto.
- Describir algunas herramientas para el modelado de tareas existentes en la actualidad.
- Realizar una herramienta colabarotiva para la especificación y el modelado de tareas permitiendo que distintos usuarios la puedan utilizar de manera conjunta sin tener que encontrarse físicamente en el mismo lugar.
- Dar ideas sobre las mejoras que se le podrían aplicar a la herramienta en un futuro.

1.5. ESTRUCTURA DE LA MEMORIA

Una descripción detallada de cómo esta estructurado este documento de Proyecto Final de Carrera es la siguiente:

En el **primer Capítulo** se realiza una justificación de la realización de este proyecto que no es otra que la necesidad de incorporar factores de calidad en el desarrollo de herramientas software más concretamente en las que se dedican al modelado de tareas. Uno de estos factores es el que permitan el trabajo en grupo ya que las herramientas existentes están enfocadas para el trabajo monousuario y en los últimos tiempos el hábito de trabajo es multiusuario.

En el **Capítulo dos** se realiza un estudio sobre la forma de trabajo actual, destacando la tendencia hacia una forma de trabajo grupal. Se describen conceptos relacionados con este término como CSCW y Groupware y cómo se pueden aplicar estos conceptos en la especificación y modelado de tareas, realizando una visión de las técnicas de modelado de tareas existentes en la actualidad.

En el **Capítulo tres** se realiza un recorrido por las distintas herramientas que soportan el modelado de tareas, y más concretamente las herramientas CTTE

(Paternó 1999) e IdealXML (Montero 2005) que servirán de base para la construcción de la herramienta propuesta en este Proyecto Final de Carrera; CoCTT.

En el **Capítulo cuatro** se presenta CoCTT, la herramienta que se ha elaborado como culminación a los estudios realizados para este proyecto. Se muestran los detalles de la implementación así como los requisitos y los correspondientes diagramas que se han utilizado para llevarla a cabo. La herramienta reúne todos los conceptos detallados en los capítulos anteriores y da soporte al trabajo de distintos individuos en un entorno colaborativo.

En el **Capítulo cinco** se describe un caso de estudio para ver como se comportaría la herramienta en una situación real. Se realizan ejemplos y se comprueba el grado de dificultad con que los usuarios la manejan.

El **Capítulo seis** recoge las conclusiones, reflexiones y resultados producidos en este proyecto así como las posibles mejoras que se podrían realizar en un futuro sobre la herramienta.

Capítulo 2

Estado del arte

2.1. INTRODUCCIÓN

En el desarrollo de una aplicación software es raro la vez que se trabaja de forma individual. Lo normal es que las personas se repartan el trabajo, especializándose cada una en una tarea determinada.

En los proyectos que son llevados a cabo por una única persona surgen problemas tales como:

- Aislamiento.
- Falta de comunicación y de contraste de ideas.
- Falta de especialización, una persona única rara vez es conocedora de todas las áreas implicadas en la construcción de un proyecto software.
- Pobreza en el desarrollo debido a que se pierde mucho tiempo en estudiar el problema e investigar posibles soluciones.
- Posibles retrasos en el proyecto.

Sin embargo los proyectos en grupo permiten simultanear tareas pudiendo ser repartidas entre los distintos miembros. Esto puede evitar muchos de los problemas citados anteriormente como la falta de especialización o la dependencia del proyecto de una única persona. Aún así, nos encontramos con otros problemas que tendrán que ser superados para poder llevar a cabo tareas colaborativas:

- Movilidad y disponibilidad de los miembros del grupo: El trabajo depende de varias personas y hay que enfrentarse a los problemas de la no disponibilidad. Estos hechos pueden llevar consigo un aumento en la duración del trabajo realizado.
- Desplazamientos y reuniones son indispensables para tener un conocimiento completo y comprobar la evolución del proyecto en todo momento, así como compartir nuevas ideas y conocimientos con el resto de miembros del grupo.
- Coordinación e integración de las tareas realizadas de forma individual por cada miembro y que al final tienen que ser puestas en común.

- Falta de comunicación, debido a que cada miembro desconoce la labor del otro y esto puede dificultar más el proceso de integración.

Como se puede ver existen pros y contras a la hora de realizar el trabajo en grupo, pero si se consigue solventar los problemas que este acarrea se conseguirá un aumento tanto en la calidad de trabajo desarrollado como en la satisfacción final del usuario.

2.2. DINÁMICAS DEL TRABAJO EN GRUPO

Se puede decir que un grupo es un conjunto de individuos que regularmente interactúan entre sí y se consideran interdependientes en relación al logro de una o más metas.

Cuando las personas trabajan juntas como equipo, cada una se beneficia del conocimiento, trabajo y apoyo de los demás miembros, lo cual lleva a una mayor productividad que la que sería lograda por una única persona que trabajara al máximo de su capacidad.

Desde la más impresionante obra de ingeniería hasta una acción tan cotidiana como dar a un interruptor, entraña un complejo entramado de acciones coordinadas por un grupo de personas. De igual modo, ocurre en el desarrollo de una aplicación software (véase Figura 2-1).

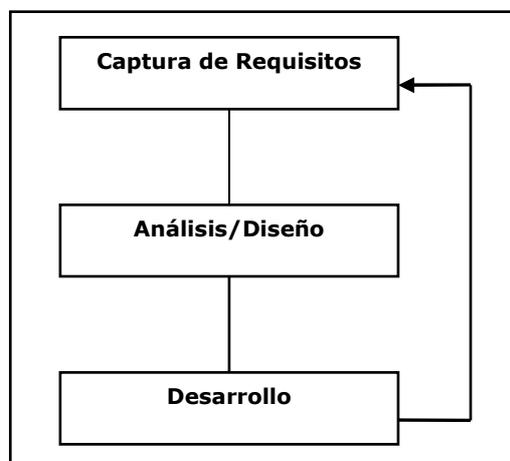


Figura 2-1. Posible ciclo de desarrollo de una aplicación tradicional

Cuando el desarrollo lo realiza una única persona, el ciclo es bastante simple. Este ciclo puede resultar adecuado para trabajos de mínima complejidad pero no apto para trabajos de gran envergadura.

En el caso de que el desarrollo sea realizado por un grupo de personas, las tareas se pueden simultanear, lo que proporciona mayor agilidad al proceso, teniendo como contrapartida la necesidad de integración. El ciclo de desarrollo de una aplicación multiusuario se muestra en la Figura 2-2.

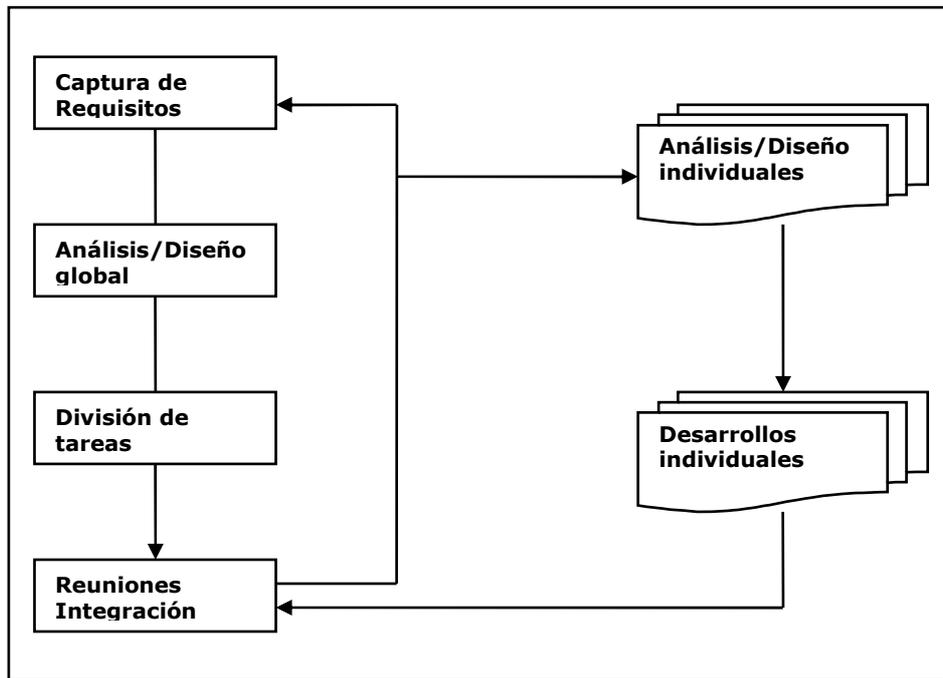


Figura 2-2. Ciclo de desarrollo de una aplicación multiusuario

Rational Unified Process (RUP), al ser una metodología genérica, permite modelar sistemas colaborativos, desde la recogida de requisitos hasta la integración del sistema, como se aprecia en la figura anterior, sin embargo, se podría enriquecer esta metodología si tuviese en cuenta la forma en la que se comunican, colaboran y se coordinan los individuos. El trabajo colaborativo (Groupware) permite la interacción entre usuarios y el envío de información de manera rápida entre miembros del equipo, para esto, las soluciones colaborativas proveen tres funciones principales [DAV1988], Figura 2-3.



Figura 2-3. Principales funciones del trabajo colaborativo (DAV, 1998)

La *comunicación* es la cualidad humana que permite el intercambio de información entre personas. Es importante que la comunicación sea eficaz, es decir, que quien envía y recibe la información perciban el mismo concepto y eficiente en cuanto al consumo mínimo de recursos. En el proceso de comunicación se identifican distintos elementos que se muestran en la Figura 2-4: Participantes (emisor y receptor), mensaje y medio.

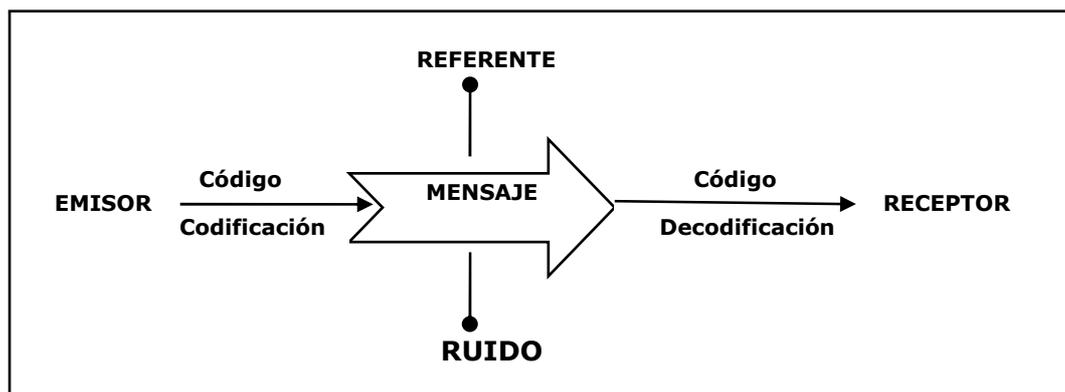


Figura 2-4. Elementos del proceso de comunicación

De igual forma, en un sistema informático es posible reconocer cada elemento: La información está contenida en documentos, los protocolos de interacción posibilitan el intercambio y el medio por el que se comunica puede variar

dependiendo de las condiciones de la interacción, comunicación cara a cara, a través de Chat, mail, fax, etc.

La *colaboración* implica la participación intencionada y coordinada de los miembros de un grupo. Se han de identificar los actores de los grupos encargados de realizar las tareas de acuerdo a sus capacidades. Proporciona la ventaja de resolver los problemas de las asambleas tradicionales como lugar y tiempo además de mejorar la eficiencia de las comunicaciones con la contribución de todos los miembros del grupo.

La *coordinación* es una actividad orientada a gestionar las dependencias entre actividades realizadas en grupo para alcanzar un objetivo y asegurar que el equipo esta trabajando de forma eficiente y en conjunto para alcanzar una meta.

La realización de una aplicación colaborativa conlleva una complejidad adicional, pues se tiene que tener en cuenta además de aspectos convencionales que puedan poseer el resto de herramientas, aspectos relacionados con el usuario como miembro de un grupo como los que se acaban de comentar. El usuario ya no se enfrenta solo al manejo de la aplicación, sino que la comparte con otros usuarios que no tienen por qué estar físicamente en el mismo lugar que él. Esto conlleva una nueva forma de actuar frente al sistema, debiendo tener en cuenta ciertos factores a la hora de realizar una aplicación colaborativa, pues ellos son los que la van a diferenciar de un sistema monousuario.

Se puede decir que los factores que más destacan de los grupos son los siguientes, (véase Figura 2-5):

- *Roles*: Conjunto de patrones de comportamiento que se esperan de alguien.
- *Normas*: Estándares aceptables que condicionan el comportamiento del grupo y de sus miembros. Son específicas para cada grupo.
- *Conformidad*: Impacto de las presiones del grupo sobre la actitud y juicio de un miembro.
- *Status*: Grado de prestigio o posición dentro de un determinado grupo.
- *Tamaño*: Número de personas que forman el grupo. Los grupos pequeños son más rápidos, los grandes generan más alternativas. En general se puede decir que a mayor tamaño del grupo menor productividad individual pues se diluye la responsabilidad.

- *Cohesión*: Fuerza del grupo marcada por el grado de cooperación y acuerdos que haya dentro del grupo.

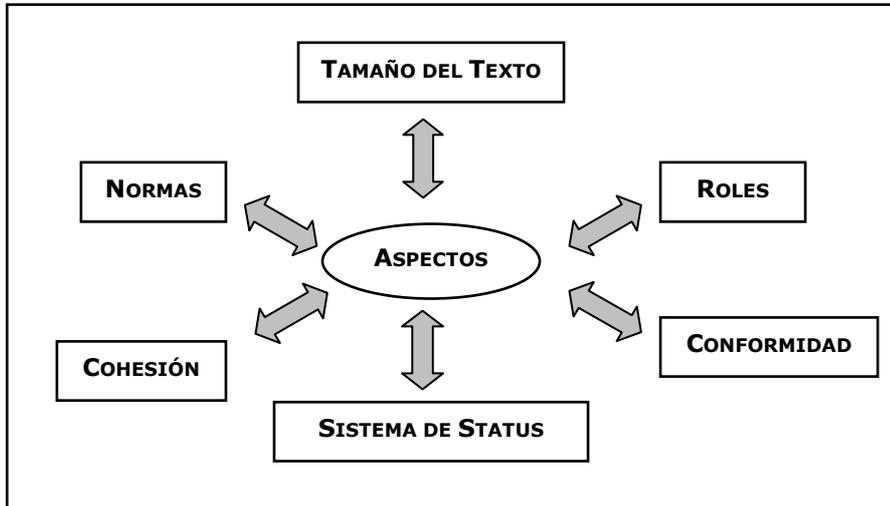


Figura 2-5. Aspectos más relevantes de un grupo

Para mejorar las relaciones entre los integrantes de un equipo que tiene que realizar un trabajo conjunto así como facilitar el desarrollo de este, existen ciertas dinámicas; Delphi, Brainstorming, Cognitive Maps, Círculos de calidad, Juegos colaborativos, Foro, Entrevista colectiva, Mesa redonda, el regalo de la alegría, Discusión dirigida, phillips 6/6 ó técnica de grupo nominal son algunas de ellas.

Delphi: Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. Se realiza un interrogatorio a expertos, con la ayuda de cuestionarios sucesivos con el fin de poner de manifiesto convergencia de opiniones.

La estrategia que sigue Delphi es la siguiente:

- Se inicia con un cuestionario.
- Se analizan las respuestas y se construye una lista según el tema.
- Se realiza un segundo cuestionario recogiendo opiniones y votos.
- Se analizan los votos y se totaliza.
- Se hace un tercer cuestionario donde se identifican los acuerdos y desacuerdos entre los participantes.

- Por último se realiza el informe final.

Brainstorming: Es una técnica en la que los participantes expresan con absoluta libertad todo lo que se les ocurra a propósito de un tema o como solución a un problema, (véase Figura 2-6). Sin ningún análisis ni filtro sobre su calidad, se anotan en la pizarra. Sólo al final, cuando se agotan la producción de ideas, se realiza una evaluación de las mismas. Es una técnica que permite desarrollar la creatividad y se utiliza para cubrir aspectos nuevos, resolver problemas o superar el conformismo y la monotonía. Antes de comenzar la tormenta se expone el problema y se explican las reglas:

- Las ideas se expresan con independencia de su calidad; no se valorará ninguna idea hasta que se diga la última frase.
- Se recomienda asociar libremente las ideas propias con las ya expuestas. cuantas más intervenciones, más posibilidades de encontrar posibilidades válidas.
- Los turnos de palabra se concederán de manera indiscriminada.

Al final, tres o cuatro personas que no hayan participado en la fase de producción analizarán todas las ideas para valorar su utilidad en función del objetivo que se pretendía con el empleo de la técnica.



Figura 2-6. Dinámica de grupo: Brainstorming

Cognitive Maps: Utilizan estructuras en términos de nodos con enlaces que representan relaciones, permitiendo observar los factores que son relevantes y las relaciones entre factores. La utilización de esta estrategia para fortalecer el trabajo colaborativo permite conocer la posición individual de los miembros del grupo frente a un objetivo común, de tal forma que cada nodo representa una idea y los nodos se van relacionando entre sí identificando posiciones en contra y a favor.

- Se construye una representación gráfica de las posiciones de las personas.
- Cada idea representa un nodo en forma de árbol.
- Las ideas relacionadas se unen así como las posiciones a favor y en contra.

Círculos de calidad: Un círculo de calidad es un grupo voluntario compuesto de trabajadores que se reúnen para mejorar el ritmo y la calidad de lo que hacen, y presentar dichas mejoras a la dirección (Figura 2-7). Algunas de las áreas que abarcan son la mejora de la seguridad, el diseño de productos y la mejora en los procesos de producción. Una de las ventajas de los círculos de calidad es la permanencia, ya que los grupos permanecen entre proyecto y proyecto. En los círculos de calidad existe un líder que propone una agenda y administra la reunión y al final, se definen tareas y responsabilidades. Su principal objetivo es identificar los problemas y proponer soluciones.



Figura 2-7. Dinámica de grupo: círculos de calidad

Juegos Colaborativos: Son entornos interactivos en modo texto y gráfico que permiten reunir personas en un mundo imaginario donde pueden interactuar de formas diversas y donde todos pueden manipular su entorno. Apoyan el aprendizaje y promueven la comunicación de una manera más natural. Los modelos para trabajo colaborativo con soporte multimedia pretenden describir un ambiente de aprendizaje colaborativo basado en servicios de comunicaciones síncronos y asíncronos. La estrategia colaborativa depende del tipo de servicio que ofrece y de la tarea a realizar. Un ejemplo de este tipo de juegos es Second Life, (véase Figura 2-8), donde jugadores virtuales se transforman en personajes que conviven en un mundo imaginario.



Figura 2-8. Juego colaborativo Second Life

Foro: El grupo en su totalidad discute informalmente un tema o problema conducido por un coordinador ante un auditorio, (Figura 2-9). La finalidad del Foro es permitir la libre expresión de ideas y opiniones a todos los integrantes de un grupo, en un clima informal de mínimas limitaciones. En razón de esta circunstancia, el coordinador o moderador del Foro juega un papel muy importante, pues debe controlar la participación espontánea, imprevisible, heterogénea de un público a veces numeroso y desconocido. Un secretario o ayudante puede colaborar con el moderador, y observar o anotar por orden a quienes solicitan la palabra, así como también para hacer el resumen de las ideas expresadas y colaborar con el coordinador para elaborar la síntesis final. Esta técnica es una de las más utilizadas debido a que trae numerosas ventajas de las cuales se pueden destacar las siguientes:

- Permite la discusión y participación.
- Permite la libre exposición de ideas y opiniones de los miembros del grupo y esto es posible de una manera informal y con pocas limitaciones.
- Da oportunidad de conocer las opiniones del grupo sobre el tema tratado.
- El auditorio puede reflexionar también sobre el tema.



Figura 2-9. Dinámica de grupo: foro

Entrevista colectiva: Un equipo de expertos elegidos por el grupo interroga a un experto ante el auditorio sobre un tema de interés, (véase Figura 2-10).

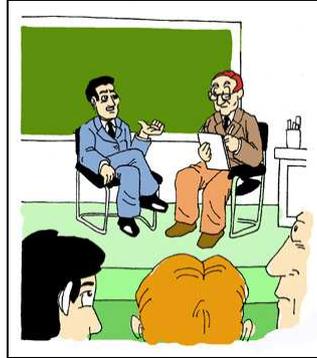


Figura 2-10. Dinámica de grupo: entrevista

Mesa Redonda: Un grupo de expertos que satisfacen puntos de vista divergentes o contradictorios sobre un mismo tema exponen ante el grupo de forma sucesiva, como se muestra en la Figura 2-11. Se utiliza esta técnica cuando se desea dar a conocer a un auditorio los puntos de vista divergentes o contradictorios de varios especialistas sobre un determinada tema o cuestión. La Mesa redonda ha sido difundida ampliamente por la televisión, donde, por ejemplo, políticos del partido gobernante y otros de la oposición, exponen sus puntos de vista contradictorios acerca de un hecho o medida de gobierno, las concepciones ideológicas, etc. Algunas de sus ventajas son las siguientes:

- La confrontación de enfoques y puntos de vista permitirá al auditorio obtener una información variada y ecuánime sobre el asunto que se trate, evitándose así los enfoques parciales, unilaterales o tendenciosos, posibles en toda conferencia unipersonal.
- La Mesa redonda tiene un director o coordinador cuya función principal al final es sintetizar los aspectos de coincidencia y los de divergencia para que el público forme su propio criterio. En cuanto a la duración, es conveniente que se extienda más allá de los 50 minutos, para permitir luego las preguntas que desee formular el auditorio durante un lapso de tiempo que se considere prudencial.



Figura 2-11. Dinámica de grupo: mesa redonda

El regalo de la alegría: Su objetivo es promover un clima de confianza personal, de valoración de las personas y de estímulo positivo del grupo. Dar y recibir un feedback positivo en ambiente grupal.

Discusión dirigida: Consiste en un intercambio de ideas entre varios participantes que han trabajado sobre un tema que puede analizarse desde distintas posiciones. Esta técnica se centra en profundizar en los conocimientos mediante un análisis crítico de los temas y estimular la comunicación interpersonal, la tolerancia y el trabajo en equipo y suele utilizarse de apoyo a otras técnicas de grupo cuando por alguna razón se necesite: promover rápidamente la participación de todo el grupo, obtener muchas opiniones en poco tiempo, resolver un problema de forma creativa o descubrir las divergencias existentes ante un tema en concreto. Una ilustración de esta dinámica se muestra en la Figura 2-12.

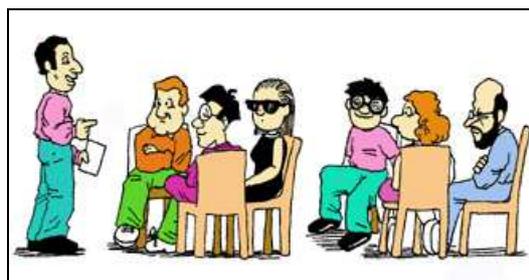


Figura 2-12. Dinámica de grupo: discusión dirigida

Phillips 6/6: Esta técnica consiste en dividir el grupo en subgrupos de, como máximo, seis componentes que durante seis minutos discutirán para responder a una pregunta o resolver un problema o caso formulado por el moderador. Para ello, se debe tener en cuenta el siguiente procedimiento: Los grupos se reúnen en salas diferentes y cada uno de los integrantes expone su opinión durante un minuto. Un

secretario designado por el grupo tomará nota de las aportaciones, y en el último minuto, se realiza un resumen de opinión del subgrupo. Un portavoz de cada grupo expone en el aula común sus resultados, que una vez comparados con los del resto de los subgrupos serán sintetizados por el moderador y anotados en la pizarra. Si todavía quedasen puntos por tratar se repite el proceso hasta que se hayan trabajado todos los aspectos, véase Figura 2-13.



Figura 2-13. Dinámica de grupo: philips 6/6

Técnica de grupo nominal: Los miembros del grupo interactúan muy poco y aportan sus decisiones de manera individual sumando después sus resultados y utilizando la votación como medio de conseguir una valoración grupal. Con más de doce integrantes se trabajará en subgrupos para seleccionar unas ideas antes de continuar con el grupo completo. El objetivo de esta técnica es intercambiar informaciones, toma de decisiones en común, lograr un alto grado de consenso, equilibrar el alto grado de participación entre los miembros del grupo y obtener una idea clara de las opiniones de los miembros.

El tiempo y la experiencia han contribuido a hacer de estas técnicas altamente eficaces a la hora de trabajar en grupo pero como se observa, todas ellas requieren una comunicación fluida de los integrantes y esto puede suponer un problema si estos no se encuentran en el mismo lugar.

Hasta la fecha y sobre todo gracias al gran avance que ha tenido Internet en los últimos años, los miembros de un equipo pueden comunicarse a través de Chat, mail, foros, pizarras...etc. ó a través de otros medios más convencionales como teléfono, fax, cartas, pero el problema aparece cuando se quiere realizar un trabajo conjunto que requiere una gran cantidad de tiempo. Cada uno por separado tendría que elaborar la tarea asignada para posteriormente ponerlas todas en común. Es

evidente que la efectividad de tareas relativas a la coordinación, colaboración y comunicación disminuiría pues cada usuario sería conocedor de su propio trabajo realizado, pero no del trabajo de los demás. La eficacia y calidad del trabajo llevado a cabo por los miembros de un equipo aumentaría si cada uno de ellos pudiese ver el trabajo realizado por los otros, permitiendo el contraste de opiniones, resolución de dudas, seguridad, cercanía y uno de los factores más importantes hoy en día, el ahorro de tiempo. Es por ello que nociones como CSCW y Groupware están cobrando cada vez más importancia en el mundo en el que nos encontramos.

2.3. CONCEPTOS CSCW Y GROUPWARE

La creciente implantación de las nuevas tecnologías en los diferentes campos que abarcan universidades, centros de investigación, empresas... etc. está haciendo que aparezcan nuevos términos para designar al fenómeno producido por los cambios. A continuación se muestran algunos de ellos que ayudarán a comprender, si cabe un poco mejor, el concepto de trabajo colaborativo.

2.3.1. Trabajo cooperativo soportado por computador

Los sistemas que soportan el trabajo en grupo en entornos de trabajo compartidos se desarrollan bajo el paradigma denominado trabajo cooperativo soportado por computadores (en inglés Computer Support Cooperative Work o CSCW).

El término CSCW fue definido originalmente por Irene Greif y Paul Cashman en 1984 como "una vía para describir cómo la tecnología de los computadores puede ayudar a los usuarios a trabajar conjuntamente en grupos" [Bannon et.al., 1989].

Esta se podría decir que es la definición más formal, pero se pueden encontrar otras definiciones relativas al CSCW:

"CSCW a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques" [Wilson, 1991].

“Sistemas basados en computadores que apoyan a un grupo de personas que trabajan en una tarea o meta común y que aprovecha una interfaz o un ambiente compartido” [Ellis, Gibas, Rein 91].

Desde el comienzo de los primeros computadores se tuvo la necesidad de compartir recursos de cómputo, como la memoria, las unidades de almacenamiento y principalmente el procesador, pero no se compartía la información ya que había diferentes barreras que lo impedían, como las distancias entre distintos lugares, las limitaciones de las comunicaciones o la diferencia de sistemas operativos.

El hecho de que los sistemas informáticos sean cada vez más complejos y principalmente la llegada de Internet, ha hecho que algunas de las barreras que existían en un principio se rompieran favoreciendo el desarrollo de CSCW. El incremento de la velocidad de procesamiento de datos, el aumento del poder computacional, la disminución de costos o el aumento de ancho de banda de las redes han dado lugar a un rápido crecimiento del CSCW. La Figura 2-14 muestra cómo han ido evolucionando los sistemas software desde 1965.

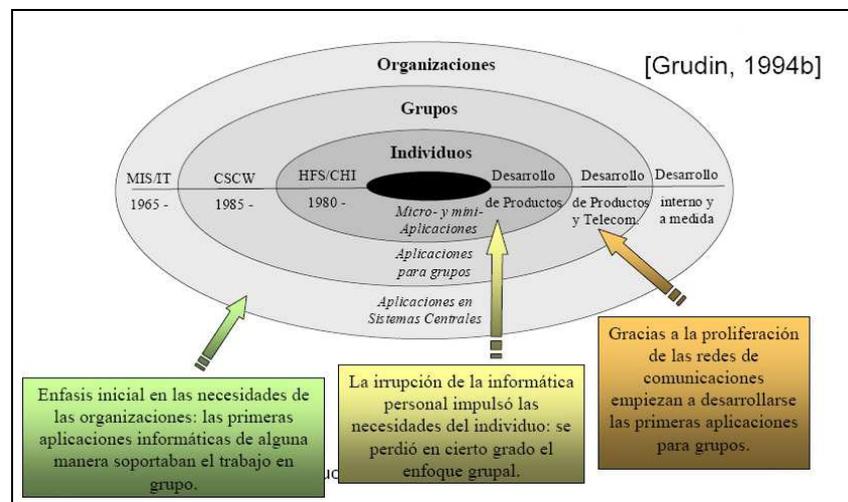


Figura 2-14. Evolución de los sistemas software [Grudin, 1994]

Es interesante destacar que en los inicios del CSCW colaboraron un número heterogéneo de personas con conocimientos en distintas áreas como la psicología, la sociología o la antropología, además de personas con conocimientos en el campo de la informática. Estos campos, fueron y son la base para el desarrollo de aplicaciones destinadas a la interacción de grupos de personas.

Entre las características más importantes del CSCW cabe destacar las siguientes:

Comunicación entre miembros del grupo: Las reuniones presenciales tienen una componente de audio y otra de video y permiten visualizar mucha información. La comunicación a través del computador está muchas veces limitada a ciertos canales donde el usuario puede leer mensajes escritos.

Se comparte la información: Es importante para prevenir la duplicación innecesaria del esfuerzo y asegurar que todos los miembros estén utilizando la misma información.

Coordinación y control de objetos compartidos: La coordinación en el sentido de que si se produce un cambio o una modificación este ha de ser visualizado por el resto de miembros del grupo.

Compartición de un espacio de trabajo: Los miembros de un grupo necesitan un espacio de trabajo común donde desarrollar sus ideas.

Organización y entendimiento común del proceso de trabajo: Las personas que trabajan conjuntamente tienen que saber qué es lo que se quiere hacer y cómo.

Soporte de toma de decisiones: Las decisiones pueden estar relacionadas con los objetivos de una tarea común, el método de trabajo que ha de adoptar el grupo, la elección de los miembros del grupo, etc.

El CSCW supuso un gran avance tecnológico logrando compartir información pero sin una herramienta adecuada no se pueden realizar las actividades de colaboración. A partir de esta necesidad se comenzó a trabajar en un nuevo tipo de tecnología: software colaborativo. De este modo surgió el concepto de Groupware.

2.3.2. Groupware

Fue acuñado por primera vez por Peter y Trdy Jonson-Lenz en 1978 del siguiente modo: "Internacional group processes and procedures to achieve specific purposes plus software tools designed to support and facilitate the group`s work"

[Jonson-Lenz, 1981], pero al igual que el concepto de CSCW se le han otorgado otras definiciones, algunas de las cuales se enuncian a continuación:

“Sistemas computarizados que ayudan a grupos de gente que se comprometen en una tarea o meta común y que proporcionan una interfaz de ambiente compartido” [Ellis, 1993].

“Colaboración mediante ordenador que incrementa la productividad o funcionalidad de los procesos entre las personas” [Coleman,1995].

“Sistema de herramientas lógicas para facilitar la cooperación de las personas en el trabajo” [Douglas C. Engelbart, 1988].

“Software que ayuda a los grupos de personas a comunicarse electrónicamente” [Gol, 1994].

“Sistemas basados en computadores que apoyan a grupos de personas que trabajan en una tarea común y que proveen una interfaz para un ambiente compartido” [Dave Chaffney].

“Colección de tecnologías que nos permiten representar procesos complejos centrados en las actividades cognitivas humanas, o bien, elementos de software que permiten la comunicación, colaboración y cooperación efectiva en un grupo de agentes activos distribuidos que trabajan de manera coordinada” [Ayala,1997].

“Tecnología de la información utilizada para ayudar a las personas a trabajar en grupo de manera eficiente” [Malone, 1992].

El Groupware es el software y hardware que soporta y ayuda al trabajo en grupo. No sirve para eliminar otros tipos de comunicación sino para añadir una nueva componente en el proceso de colaboración aumentando la eficacia de trabajo en tres niveles clave, que como se ha comentado anteriormente, dan soporte a la interacción grupal: Comunicación, colaboración y coordinación.

De entre las características más relevantes del Groupware se pueden destacar tres fundamentalmente:

- i. La información debe mantenerse en un sólo sitio común a todos los miembros.
- ii. Debe permitir la interacción entre usuarios ya sea de forma escrita, voz o video.
- iii. Debe proveer un ambiente de colaboración en el que se perciba que el trabajo en grupo se lleva a cabo.

En el desarrollo y la evolución del Groupware, el campo de la Informática ha aportado conocimientos de gran valor [Baec,1993]. La Interacción persona-ordenador ha contribuido con el análisis y diseño de la interacción entre el usuario y el sistema. Los Sistemas Operativos y las Bases de Datos han contribuido en la administración de recursos y control de concurrencia de procesos. El campo de las redes y las comunicaciones han aportado conocimientos sobre sistemas distribuidos y arquitectura cliente-servidor. La Inteligencia Artificial ha permitido el desarrollo de agentes de apoyo y planificación del usuario así como la coordinación y la negociación en el grupo. Por último, el campo de la multimedia proporciona un soporte de audio y video que hace posible la utilización de servicios de videoconferencia.

Actualmente, se tienen muchas clasificaciones de Groupware, algunas de centran en la funcionalidad, otras en el grado de sistemas que soportan el trabajo en grupo o en términos de espacio tiempo. A continuación, se presentan las más características.

2.3.2.1. De acuerdo al espacio / tiempo de los miembros del grupo

Esta taxonomía (recogida en la la Tabla 2-1 y la Figura 2-15) es debida a Robert Johansen y considera las aplicaciones Groupware de acuerdo al tipo de interacción de los miembros del grupo de trabajo. Algunos ejemplos de aplicaciones Groupware se muestran en la Tabla 2-2.

En cuanto a tiempo:

- **Interacción sincronía:** Herramientas que facilitan la comunicación en un mismo lugar (espacio) y un mismo momento (tiempo).
- **Interacción Asíncrona:** Herramientas que sirven para facilitar los procesos de comunicación y permiten compartir información entre personas que no interactúan simultáneamente.

En cuanto a espacio:

- **Local:** En donde el manejo de clientes y servidores se realiza en el mismo espacio.
- **Distribuida:** Donde el manejo de clientes y servidores se realiza en diferentes espacios

	Mismo Tiempo	Diferente Tiempo
Mismo lugar	Interacción síncrona	Interacción asíncrona
Diferente Lugar	Interacción distribuida síncrona	Interacción distribuida asíncrona

Tabla 2-1 Taxonomía espacio-temporal del groupware

	Síncrono (mismo tiempo)	Asíncrono (distinto tiempo)
Local (mismo lugar)	Reuniones cara a cara Utilidades con respuesta de la audiencia. Entornos de conversación y tormenta de ideas. Pantallas compartidas para aplicaciones.	Administración / Manejo de Datos Raramente utilizado. Un ejemplo: Trabajo en turnos (en el mismo ordenador).
Remoto (lugares distintos)	Reuniones Remotas Pizarra electrónica Charla (Chat) Aplicaciones compartidas Vides / tele conferencia	Mecanismos de Coordinación Transferencia de ficheros Correo electrónico Foros de debate Flujo de trabajo (Workflow)

Tabla 2-2. Clasificación del groupware [Johansen, 1988; Ortega, 2001]

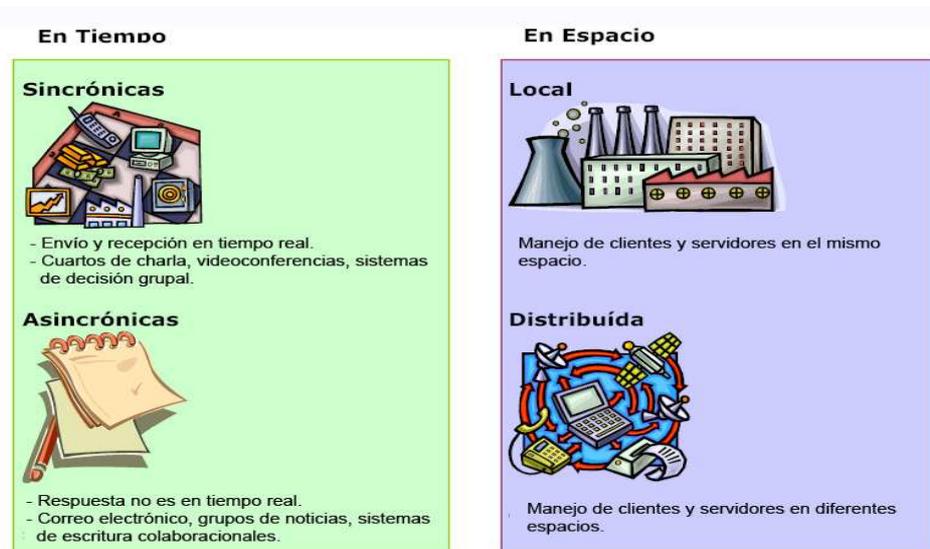


Figura 2-15. Clasificación de herramientas colaborativas en función del espacio/tiempo

2.3.2.2. De acuerdo al manejo de información

Dependiendo del soporte que brindan los sistemas Groupware se puede clasificar en:

- **Sistemas para compartir información:** Manejan la información de la organización y apoyan actividades individuales de un conjunto de usuarios que comparten información.
- **Sistemas cooperativos:** Están caracterizados por manejar información personal y de la organización, apoyando las actividades de un grupo de trabajo determinado.
- **Sistemas concurrentes:** Dan soporte al trabajo en grupo al igual que los sistemas cooperativos pero la información que manejan es procesada por múltiples usuarios de manera concurrente.

2.3.2.3. De acuerdo al propósito de la aplicación

Dependiendo del propósito de la aplicación podemos encontrar dos tipos de aplicaciones:

- **Aplicaciones de propósito general:** Son aplicaciones comerciales que dan soporte a los trabajos en grupo. Algunos ejemplos son el "e-mail", teléfono, las agendas grupales, los sistemas para compartir archivos, los foros electrónicos, etc.
- **Aplicaciones de propósito específico:** Desarrolladas para dar apoyo a un grupo de usuarios en particular, es decir, modelan y diseñan el trabajo específico en grupo.

2.3.2.4. De acuerdo al nivel de automatización

Según algunos autores las aplicaciones Groupware se pueden clasificar de acuerdo al tipo de trabajo en grupo que apoyan: Flujo de documentos, automatización de procesos, automatización de tareas, herramientas flexibles de trabajo en grupo. Esta taxonomía se basa en la variable del Groupware que es objeto principal de

atención, ya sea el individuo, el documento o el proceso [SAA97], (Véase Tabla 2-3).

VARIABLE	DEFINICIÓN
Individuo	El sistema gestiona localmente el trabajo de un individuo dentro de un grupo
Documento	El sistema se encarga de la gestión de las taras asociadas a un documento (consulta, actualización, etc.)
Proceso	El sistema controla la finalización de las actividades

Tabla 2-3. Taxonomía de Dyson, según el objetivo principal del Groupware

2.3.2.5. De acuerdo al tipo de aplicación

Dependiendo del tipo de aplicación y funcionalidad, estos sistemas también se pueden clasificar en los siguientes grupos:

- **Sistemas de mensajes por computador:** Es el ejemplo de aplicación mas conocida de Groupware. Esta basado en el intercambio asíncrono de mensajes de texto entre grupos de usuarios. Algunos ejemplos de este tipo son el Correo Electrónico, las conferencias computacionales y los sistemas de boletines.
- **GDSSs (Group decisión support Systems):** Su objetivo es mejorar la productividad de la toma de decisiones aumentando la velocidad en la realización de este proceso. Algunos ejemplos de este tipo son: reuniones electrónicas que contienen algunas estaciones y redes de trabajo, "display" públicos controlados por computador, equipos de video y audio.
- **Sistemas de reuniones electrónicas:** Sistemas que permiten establecer discusiones remotas entre un grupo de personas.
- **Sistemas de coordinación:** Permiten a los individuos ver sus acciones, así como las acciones relevantes de otros dentro del contexto del propósito general, informando a los usuarios del estado de sus acciones.

2.3.2.6. De acuerdo a la funcionalidad de la aplicación

Según la funcionalidad de la aplicación se encuentran los siguientes sistemas Groupware:

- **Sistemas de mensajería:** Soportan el intercambio asíncrono de mensajes de texto escrito entre miembros del grupo.
- **Editores Multiusuario:** Permiten a los miembros de un grupo hacer una edición y composición conjunta del documento.
- **Sistema de soporte de decisiones grupales y salas de reuniones electrónicas:** Se pueden dividir en tres subcategorías: Las conferencias computacionales en tiempo real, la Tele conferencia computacional (video y conferencia) y la conferencia de escritorio (videoconferencia local mas Compartición de texto y gráfico).

2.3.2.7. De acuerdo al tipo de herramientas

Dependiendo de la herramienta utilizada se obtiene la siguiente clasificación de los sistemas Groupware:

- **Herramientas de comunicación-colaboración:** Herramientas de comunicación electrónica que envían mensajes, archivos, datos o documentos entre personas y facilitan la comparación de información (colaboración asíncrona), como por ejemplo: correo electrónico, correo de voz o publicación Web.
- **Herramientas de conferencia:** Facilitan la comparación de información, de forma interactiva (colaboración síncrona). Algunos ejemplos de este tipo de herramientas son las conferencias de datos, de voz o de video, salas de Chat o mensajería instantánea ó sistemas facilitar reuniones (son sistemas de conferencias integrados en una sala).
- **Herramientas de gestión colaborativa o en grupo:** Calendarios electrónicos, sistemas de gestión de proyectos para organizar y hacer un seguimiento de las acciones de un proyecto, sistemas de control de flujo de actividad para gestionar las tareas y documentos en un proceso organizado de forma estructurada o sistemas de gestión del conocimiento para recoger, organizar, gestionar y compartir varios tipos de comunicación.

A la hora de desarrollar aplicaciones Groupware se ha de tener cuidado, hay determinados factores que pueden influir negativamente en el desarrollo de este tipo de sistemas. Las aplicaciones Groupware pueden fracasar porque requieren que parte del personal realice trabajo adicional y dada la diferencia entre los

miembros del grupo (conocimientos, papeles que desempeñan, tareas que realizan...etc.) no proporcionará el mismo beneficio a todos. Debería de haber un beneficio colectivo para este tipo de aplicaciones.

Por otro lado, el Groupware también puede fallar si no se tiene en cuenta el rango de excepciones y de improvisaciones que pueden ocurrir en muchas actividades grupales ya que el desarrollo de estas suele ser particularmente variable y muchas características que son útiles en sistemas monousuario fallan al incorporarlas al sistema multiusuario.

Se puede decir que la intuición es más útil en sistemas monousuario. Por ejemplo, un director podría tener una rápida idea de la experiencia de sus empleados con una aplicación monousuario, pero fallaría al hacer lo mismo con una aplicación multiusuario.

Aún así, y dado que toda evolución y mejora siempre conllevan unos riesgos, pesa más las contribuciones positivas que producen este tipo de aplicaciones en cualquier organización. De hecho, cada vez son más los estudios realizados y las herramientas colaborativas construidas, aunque falta todavía mucha labor por hacer.

Este Proyecto Final de Carrera trata de dar un paso adelante en la construcción de una herramienta colaborativa centrada en el análisis y modelado de tareas de usuario debido a que, aunque ya existen algunas implementadas para realizar esta labor, no hay todavía ninguna o al menos no es conocida la herramienta con la que se pueda llevar a cabo estos fines, que implemente aspectos que permitan la comunicación entre usuarios.

A continuación se muestran algunos conceptos claves como *modelado de tareas*, *tipos de notaciones*...etc., que serán claves en la construcción de la aplicación final.

2.4. MODELADO DE TAREAS. NOTACIÓN CTT.

Los modelos de tareas describen las actividades lógicas realizadas por el usuario cuando éste interactúa con la aplicación representando las tareas que el usuario debe de realizar para interactuar con el sistema.

Se emplea para el análisis de requisitos de la interfaz ya que para que la aplicación sea aceptada y satisfactoria para el usuario final, la interfaz es una parte clave en el desarrollo de cualquier aplicación. El modelado de tareas para el diseño y desarrollo de interfaces de usuario consigue un nivel de abstracción superior, haciendo que el desarrollo de software se convierta en un proceso de ingeniería lo que conlleva a conseguir un software de calidad y permite obtener aplicaciones interactivas centradas en el usuario.

El proceso de análisis de tareas se compone de dos partes muy importantes; la primera es la obtención de la información necesaria para comprender las actividades que realiza el usuario (fase de análisis) y la segunda labor, es proporcionar una representación de esta información sobre un modelo adecuado (fase de modelado). Estos pasos nos darán una descripción formal del conjunto de acciones que debe realizar el usuario para conseguir un determinado objetivo. La comprensión, el conocimiento, las intenciones...etc. quedarán modelados en este análisis.

La información aportada por un modelo de tareas es útil para comprender el dominio de la aplicación identificando las actividades más importantes y sus objetivos. Permite facilitar discusiones inter disciplinares: El conocimiento de la tarea puede ser útil para el diseñador, usuarios, analistas, psicólogos, sociólogos, etc. También resulta beneficiosa a la hora de realizar el diseño de la nueva aplicación de forma consistente con el actual modelo conceptual, preservando las características más relevantes del funcionamiento lógico. Como última característica se puede destacar el beneficio aportado en el análisis y evaluación de la usabilidad.

Existen varios métodos para el análisis de tareas. Estos métodos se diferencian en el grado de formalismo y finalidad. Así, dependiendo de la finalidad del estudio, estos modelos pueden ser clasificados en tres grandes grupos: cognitivos, predictivos y descriptivos. Los *métodos cognitivos* representan el tipo de conocimiento que debe tener un usuario identificando el comportamiento correcto que debe tener con el sistema. Los *métodos predictivos* describen secuencias de comportamiento que debe tener el usuario con el sistema y por último, los *métodos descriptivos*, permiten hacer una descripción del sistema a partir de la información obtenida de las tareas.

El modelado de tareas permite describir las relaciones entre varias tareas identificándolas. Estos modelos se componen de tres elementos fundamentales: Objetivos, Tareas y Acciones. Las *tareas* son las actividades que el usuario debe realizar para alcanzar un objetivo, un *objetivo* es un estado o logro que el usuario quiere alcanzar dentro de una aplicación y una *acción* identifica cada uno de los pasos a seguir para cumplimentar una tarea.

Los modelos de tareas se estructuran en diferentes niveles de abstracción. El grado de descomposición será distinto en función del propósito perseguido. De este modo y dependiendo de la finalidad que se desee conseguir, los modelos de tareas permiten:

- Comprender mejor el dominio de la aplicación, pues se identifica de forma precisa las principales actividades y sus relaciones.
- Poseer una naturaleza interdisciplinar debido a que en el diseño intervienen personas de distinto ámbito: diseñadores, usuarios, expertos...etc.
- Contar con una herramienta automática que permita poder realizar el modelado de las tareas.
- Formar parte de la documentación de un sistema interactivo debido a su naturaleza gráfica facilitando el entendimiento tanto para los usuarios como para los diseñadores, de cara a facilitar la comunicación entre ambos.
- Contar con una estructura jerárquica, que permita manejar distintos niveles de abstracción y de lugar a un refinamiento progresivo del análisis de tareas.
- Facilidad para expresar relaciones temporales entre las tareas.
- Analizar la usabilidad de los sistemas interactivos

Para conseguir obtener un buen análisis se deben de identificar las tareas más importantes del sistema. Esta información se puede conseguir de diferentes formas; mediante entrevistas y reuniones, cuestionarios, observación del comportamiento de los propios usuarios, analizando el entorno, así como revisando la documentación actual, programas...etc. Se debe obtener información sobre qué necesita el usuario para realizar una tarea, la terminología y símbolos del dominio del problema, una descripción de cómo esas tareas se realizan actualmente, así como casos de uso y tipos de usuarios que las llevarán a cabo.

De este análisis se obtendrán las tareas más importantes con información adicional como atributos, restricciones, preferencias... etc. De esta información, se puede obtener la información más relevante como el *modelo de diálogo*; cómo se va a realizar la comunicación persona-ordenador, bajo qué paradigma y estilo, el *modelo de tareas*; especificación de las tareas en el nuevo sistema, el *dominio de sistema*; descripción de los componentes y arquitectura del sistema, el *modelo de usuarios*; identificación del tipo de usuarios, papel que desempeñan en el sistema y sus interrelaciones y las *propiedades del sistema*; estudio de las características del sistema y de los requisitos que satisface (seguridad, robustez, etc.).

Algunos ejemplos de métodos de análisis de tareas son: Hierarchical Task Analysis (HTA), Goal-Operations-Methods-Selection (GOMS), Task Action Grammar (TAG), User Action Notation (UAN) y la notación ConcurTaskTrees (CTT) que es la más reciente y en la cual se basa este Proyecto Final de Carrera.

USER ACTION NOTATION (UAN) (Hartson and Gray 1992). Es una notación de bajo nivel que permite la descripción de las acciones realizadas en el proceso de interacción entre el usuario y la máquina. Representa el comportamiento del usuario y de la interfaz cuando se realiza una tarea ya que recoge las acciones del usuario, la respuesta del sistema a estas acciones y el cambio de estado del sistema después de realizarlas. Algunas de estas acciones pueden ser: arrastrar, apuntar, pinchar en determinadas interfaces...etc. Las acciones del usuario causarán los cambios que serán visibles a través de la interfaz. UAN utiliza una tabla con tres columnas que representan las acciones del usuario, los cambios de visualización y cambios de estado de la interfaz y sus características más notables son: notación precisa, detallada y sin ambigüedades. Se caracteriza por ser un método *cognitivo y gráfico*.

LA FAMILIA GOALS, OPERATORS, METHODS, SELECTION RULES (GOMS) (Card, Moran et al. 1983). Esta técnica sirve para modelar y describir las prestaciones de las tareas desde el punto de vista humano permitiendo formalizar las actividades tanto físicas, como mentales que intervienen en el mecanismo de resolución de problemas del ser humano. Para cada tarea se describen los objetivos del usuario, definidos en un lenguaje común, los *operadores* que describen las acciones que el software permite que los usuarios lleven a cabo, los *métodos* y las *reglas* de selección las cuales utilizarán los usuarios para decidir qué

método utilizar en una determinada situación. GMOS es un método *cognitivo y textual*.

HIERARCHICAL TASK ANALYSIS (HTA) (Annett, 1967). Es la técnica de análisis de tareas más antigua. Describe un conjunto de actividades estructuradas de forma lógica en diferentes niveles y realiza una descripción de las tareas en términos de *objetivos* que los usuarios están buscando conseguir, *operaciones* que son las actividades que realizan los usuarios para alcanzar objetivos y que se descomponen jerárquicamente en subtareas del sistema y los *planes* que son las condiciones bajo las cuales las operaciones se llevan a cabo. HTA es un método *cognitivo y gráfico*.

TASK ACTION GRAMMAR (TAG) (Payne et al., 1986). Se utiliza para representar el conocimiento del usuario, describiéndolo para realizar una tarea en base a una tarea con características [HAR90]. El conocimiento del usuario se puede expresar mediante un esquema que engloba un conjunto de reglas individuales. Los esquemas están formados por una estructura sintáctica definida por un conjunto de características y éstas son atributos que definen la semántica de la tarea. Así, las tareas que tengan estructuras sintácticas diferentes tendrán un esquema diferente y se podrá medir la complejidad de la interfaz mediante la identificación de un conjunto de características en la acción de tareas y de su agrupamiento en esquemas.

CONCURTASKTREES (CTT) (Paternò, 1999). Esta notación, permite representar relaciones temporales existentes entre las tareas y entre los usuarios necesarios para llevarlas a cabo. Fue desarrollada usando la notación LOTOS, notación formal que supuso un cambio en el desarrollo de interfaces de usuario, pues permitió a los desarrolladores describir el comportamiento y los estados modificados. Sin embargo, LOTOS presentaba algunas limitaciones, ya que había una necesidad de añadir nuevas operaciones para representar de una forma más amplia el comportamiento dinámico en la interacción persona-ordenador y permitir aportar información adicional para su uso en el análisis de modelo de tareas. Además, la sintaxis de LOTOS, puede generar expresiones complicadas incluso cuando el comportamiento descrito es simple. Por todo esto, surgió la necesidad de desarrollar una nueva notación: *ConcurTaskTrees*. Esta notación puede ser usada para describir el comportamiento de aplicaciones reales con unas dimensiones de

mayor tamaño. Este es el problema de muchas otras notaciones en las que se observan a menudo limitaciones cuando se trata del estudio de casos reales.

CTT aporta un conjunto muy amplio de operadores gracias a la notación LOTOS. Estos permiten describir las relaciones temporales entre las tareas: *conurrencia*, *opcionalidad*, *activación*, *desactivación* y *interrupción* y además, para cada tarea proporciona información adicional como por ejemplo el tipo de la tarea.

Proporciona una representación gráfica en forma de árbol y muestra la descomposición jerárquica de las tareas del sistema. Una estructura jerárquica es algo muy intuitivo, de hecho, cuando la gente tiene que resolver un problema, generalmente tiende a descomponerlo en pequeños problemas manteniendo las relaciones entre varias partes de la solución. La estructura jerárquica de la notación CTT tiene una serie de ventajas: Proporciona una amplia gama de granularidad que permite que una estructura de tareas tanto grandes como pequeñas pueda ser reutilizadas, lo cual permite beneficiarse de estructuras de tareas ya realizadas para definir tanto un alto como bajo nivel. Proporciona una notación concurrente, lo que hace que se pueda definir un amplio abanico de posibles relaciones entre tareas. También cabe destacar que en la notación CTT, a diferencia de otros tipos de modelado, dos tareas pueden sincronizarse. Esto es un hecho positivo, por ejemplo cuando se necesita intercambiar información debido a que la información de salida de una tarea es la información de entrada de la tarea siguiente.

Esto hace a la notación CTT una técnica de modelado flexible, pues permite representar actividades concurrentes e iterativas donde un objetivo puede ser alcanzado realizando diferentes tareas, teniendo la posibilidad de realizar colaboraciones entre múltiples usuarios, así como una notación compacta, comprensible y representativa. El éxito clave de esta notación es la capacidad de proveer mucha información, por lo que las tareas a realizar por lo usuarios serán muy intuitivas. De este modo, la notación CTT puede ser utilizada por personas que no tengan grandes conocimientos en informática como por ejemplo los usuarios de la propia aplicación.

La realización de las tareas se indica mediante la utilización de iconos gráficos. En función del actor que las lleve a cabo podremos clasificarlas en cuatro tipos, como muestra la Tabla 2-4.

TIPO DE TAREA	NOTACIÓN GRÁFICA	DESCRIPCIÓN
Tareas de usuario		Tareas realizadas completamente por el usuario, son tareas cognitivas o físicas que no interactúan con el sistema. Describen procesos realizados por el usuario usando información que recibe del entorno.
Tareas de aplicación		Tareas realizadas por la aplicación. Pueden obtener información interna del sistema o producir información hacia el usuario.
Tareas de interacción		Son tareas que realiza el usuario interactuando con la aplicación por medio de alguna interacción. Un ejemplo puede ser elemento de una lista desplegable.
Tareas Abstractas		Requieren acciones complejas y por ello no es fácil decidir donde se van a realizar exactamente. Son tareas que serán decompuestas en un conjunto de nuevas subtareas.

Tabla 2-4. Tipos de tareas definidas en la notación CTT

Tareas del usuario: Son las tareas realizadas completamente por el usuario. Pueden ser tareas *cognitivas* o *físicas* que no interactúan con el sistema. Describen procesos realizados por el usuario usando la información que recibe del entorno, por ejemplo, seleccionar dentro de un conjunto de información la que se necesita en un instante determinado para la realización de otra tarea.

- *Tareas de Planificación:* Secuencia de actividades a realizar por el usuario.
- *Tareas de Comparación:* Cuando el usuario tiene que evaluar una determinada información, por ejemplo, comparar cantidades o identificar el máximo valor de una lista de valores.
- *Tareas de Solución del problema:* Si el usuario tiene que encontrar la solución a un problema, por ejemplo, si hay un conflicto entre la ruta de dos vuelos y el regulador tiene que encontrar modos de evitar una colisión.

Tareas de la aplicación: Son tareas activadas y realizadas por la propia aplicación. Permiten obtener información interna del sistema o producir información para el usuario. Por ejemplo, una tarea que presente los resultados obtenidos de una consulta en una base de datos.

- *Descripción:* Las aplicaciones muestran un resumen del juego de datos que no proveen los datos elementales.
- *Comparación:* Su propósito es ayudar al usuario a compara valores de datos.

- *Localización*: Las aplicaciones dan información detallada sobre los datos, lo que permite a los usuarios encontrar rápidamente la información deseada.

Tareas de interacción: Son tareas que realiza el usuario interactuando con la aplicación por medio de alguna técnica de interacción.

- *Tareas de Selección*: Son muy comunes en las aplicaciones. Por ejemplo, tareas donde el usuario debe seleccionar uno o más artículos de entre varios. La tarea se puede clasificar dependiendo de si se selecciona un único artículo o varios.
- *Tareas de Edición*: Tareas que permiten a los usuarios especificar datos de entrada y esta información puede ser modificada antes de ser definitivamente enviada a la aplicación.
- *Tareas de Control*: El propósito de estas tareas es generar un evento indicando cuando algo debe ocurrir.

Tareas abstractas: Modelan tareas cuya única finalidad es servir de agrupación de otras tareas y se pueden descomponer en otras más sencillas.

Tareas cooperativas: Son tareas que requieren la participación y cooperación de distintos usuarios para realizar actividades.

Como se ha expuesto anteriormente el conjunto de operadores de la notación CTT es una ampliación del utilizado por la notación LOTOS. Esta notación es concurrente y se utiliza para poder especificar interfaces de usuario, pues permite describir comportamientos manejados por eventos y modificaciones de estado. El conjunto de operadores de la notación CTT se muestra en la Tabla 2-5:

OPERADOR TEMPORAL	NOTACIÓN	DESCRIPCIÓN
Entrelazado (Concurrencia independiente)	T1 T2	Las acciones de las dos tareas pueden realizarse en cualquier orden.
Elección	T1 [] T2	Selección alternativa entre dos tareas. Una vez que se esta realizando una de ellas la otra no esta disponible hasta que termine la que esta activa.
Sincronización (Concurrencia con intercambio de Información)	T1 [] T2	Las dos tareas tienen que sincronizarse en alguna de sus acciones para intercambiar información.
Desactivación	T1 [> T2	Desactivar. La primera tarea es desactivada cuando comienza la ejecución de la segunda.
Activar (enabling)	T1 >> T2	Cuando termina la T1 se activa la T2. Las dos tareas se realizan de forma secuencial.
Activar con paso de información	T1 []>> T2	Cuando termina T1 genera algún valor que se pasa a T2 antes de ser activada.
Iteración	T1*	La tarea T1 se realiza de forma repetitiva. Se estará realizando hasta que otra tarea la desactive.
Independencia de Orden	T1 = T2	Ambas tareas pueden ser realizadas, pero una vez comenzada una debe finalizar antes de que comenciar con la otra.
Suspend/Resume	T1 > T2	T2 tiene la posibilidad de interrumpir a T1 que podrá ser retomada cuando aquella finalice.
Tarea Opcional	[T1]	La realización de la tarea es opcional.

Tabla 2-5. Operadores temporales definidos en la notación CTT

2.4.1. Ejemplo CTT

A modo de ejemplo de especificación utilizando este tipo de notación se muestra una parte de una aplicación para acceder a información sobre un museo. En la Figura 2-16 se muestra el modo de introducir la información por parte del usuario sobre el tipo de artista que está buscando y el sistema le presenta la información asociada a él.

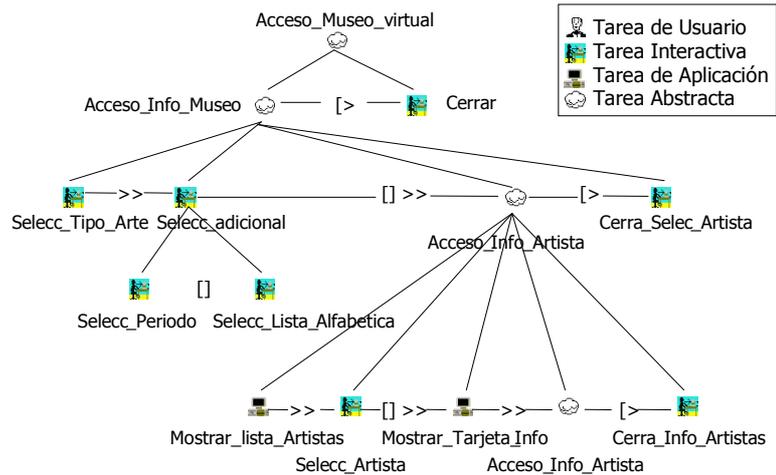


Figura 2-16 Modelado de tareas correspondiente al acceso a la información de un museo

Se comienza con la tarea "Acceso_Museo_Virtual" que puede ser interrumpida en cualquier momento ([>) por la tarea "Cerrar". En un siguiente nivel se encuentra la tarea de interacción "Selecc_Tipo_Arte" que indica el tipo de arte que el usuario seleccionará para visualizar su información, seguido (>>) por la selección del periodo histórico en el que está encuadrada la obra del artista. La siguiente tarea a realizar es el acceso a la información del artista, aunque para realizarla, se debe de tener la información de las ultimas tareas mencionadas ([]>> operador activación con paso de información). Algunas de estas tareas se van a descomponer en nuevas tareas de manera jerárquica. Así la tarea "Selecc_adicional" permite seleccionar un artista mediante una lista alfabética o mediante una lista ordenada por periodos históricos.

En las, Figura 2-17, Figura 2-18, Figura 2-19, Figura 2-20 se exponen una serie de ejemplos en los que se muestra, de forma gráfica, la utilización de los operadores temporales entre tareas.

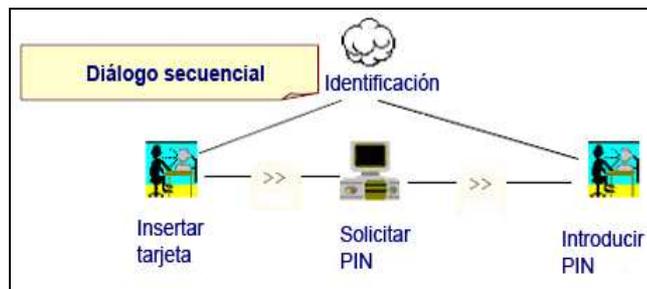


Figura 2-17. Operadores temporales: significado en el diálogo secuencial

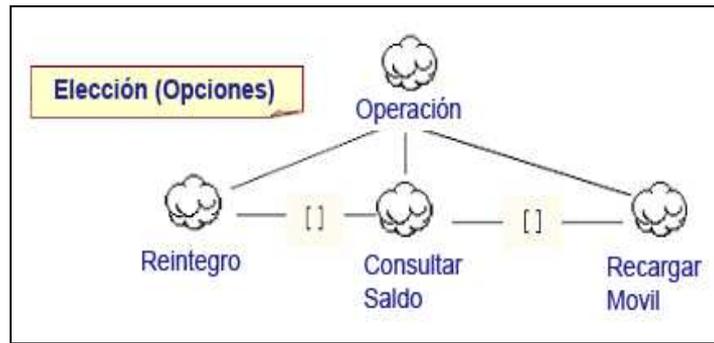


Figura 2-18. Operadores temporales: significado en el diálogo elección



Figura 2-19. Operadores temporales: significado en el diálogo concurrencia

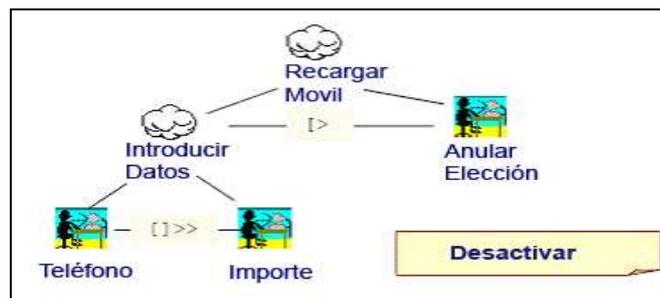


Figura 2-20. Operadores temporales: significado en el diálogo desactivar

2.5. ANÁLISIS Y CONCLUSIONES

El extenso uso que están tomando los sistemas software hace que cada vez sea mayor la gama de tareas, contextos y dispositivos en los que se deben desarrollar las aplicaciones. Poder contar con modelos, métodos y herramientas,

hace que los diseñadores puedan manejar el problema de crear aplicaciones para estos contextos tan diversos de una forma más sistemática. Poder contar con ellos, permitirá poder elegir la técnica de interacción más adecuada que permita dar soporte a las actividades a realizar por los usuarios.

Haciendo un breve recorrido por las distintas notaciones que permiten representar modelos de tareas, se ha escogido la notación CTT (Paternó 1999) por ser la que más características aporta y por su facilidad de uso, lo que hace que sea aplicable a proyectos reales con aplicaciones de un tamaño medio-largo y que conllevan especificaciones de cierta complejidad.

Para el desarrollo de una nueva herramienta con la que se puedan llevar a cabo modelos de tareas, se ha estudiado la metodología de trabajo seguida por las organizaciones actuales, llegando a la conclusión, de que las herramientas utilizadas pueden quedarse obsoletas a la hora de afrontar el trabajo en equipo. Las herramientas Groupware están tomando cada vez mayor importancia en el ámbito de construcción de sistemas software, por lo que la finalidad de este proyecto será proponer una nueva herramienta que intente subsanar las carencias actuales.

Capítulo 3

coCTT: Un entorno colaborativo para el modelado de tareas basado en CTT e IdealXML

Para la realización de este Proyecto Final de Carrera se ha optado por elegir la notación CTT para realizar el modelado de tareas pues su uso es el más extendido en la comunidad IPO y por todas las ventajas que aporta, las cuales han sido comentadas anteriormente; entorno gráfico, estructura jerárquica, posibilidad de reutilización, modelado de tareas cooperativas, pero sobretodo, por su facilidad de uso y su capacidad para poder aplicarse en proyectos reales de mediana y gran envergadura con una especificación de cierta complejidad.

3.1. HERRAMIENTAS DISPONIBLES EN EL MODELADO DE TAREAS

Aunque son pocas las herramientas de este tipo disponibles, las que hay han ayudado a agilizar el proceso de diseño de la interfaz de usuario. Para la notación KLM/GOMS existe la herramienta *Critique* (Hudson, John et al. 2000), para la notación UAN se ha desarrollado la herramienta *Quantum* (Hix and Harstson 1994) y por último y quizá sean estas las que más nos interesen, las herramientas *CTTE* (Paternò 2002) e *IdealXML* (Montero 2005) las cuales permiten a los diseñadores elaborar modelos de tareas utilizando la notación CTT.

3.2. EL EDITOR CONCURTASKTREES: CTTE

Es una herramienta gratuita desarrollada en lenguaje java que cuenta con diferentes iconos que representaran tanto las tareas, como las relaciones temporales que se pueden dar entre ellas, permitiendo un diseño jerárquico de la estructura de los modelos de tareas. Otra de las características más positivas de esta herramienta es que permite guardar los modelos de tareas creados en varios

formatos: como imagen en formato jpg, en formato LOTOS, HTML Y XML. Este último es el formato que hace más interesante el uso de esta herramienta.

La herramienta CTTE (Paternò 2002), (véase Figura 3-1), se encuentra disponible en la página Web <http://giove.cnuce.cnr.it/ctte.html>.

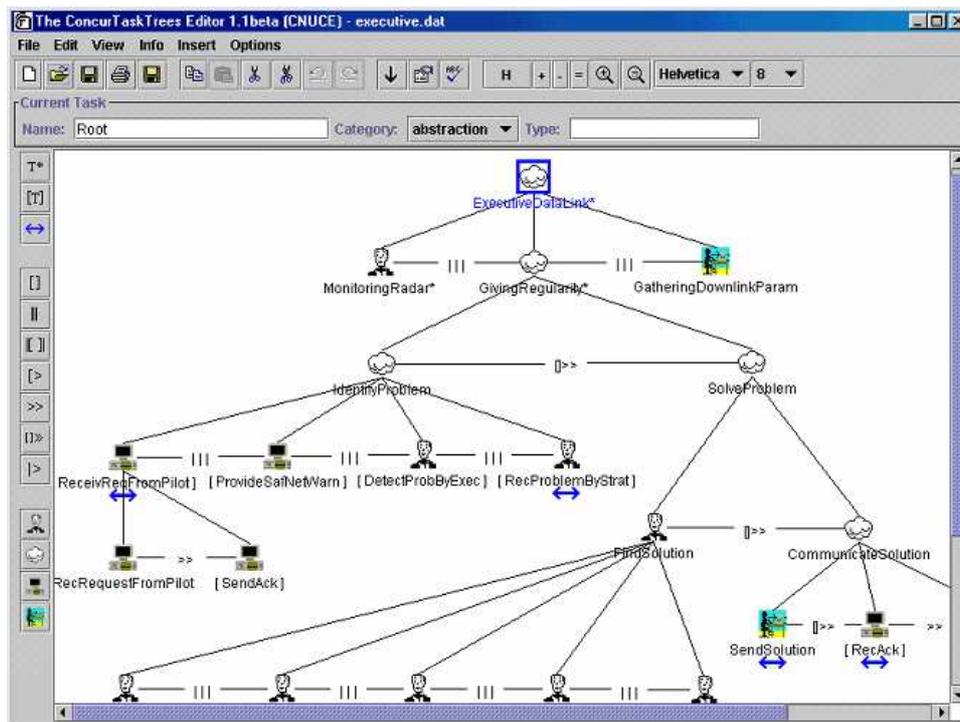


Figura 3-1. Editor CTTE

3.3. IDEALXML

IdealXML (Montero 2005), es otra herramienta que permite el modelado de tareas (véase Figura 3-2) ya que incorpora su propio editor CTT en el que, al igual que el original, se dispone de un conjunto de tareas y relaciones temporales entre ellas. La diferencia que presenta con respecto a la herramienta CTTE es el aspecto visual de los iconos asociados a las tareas.

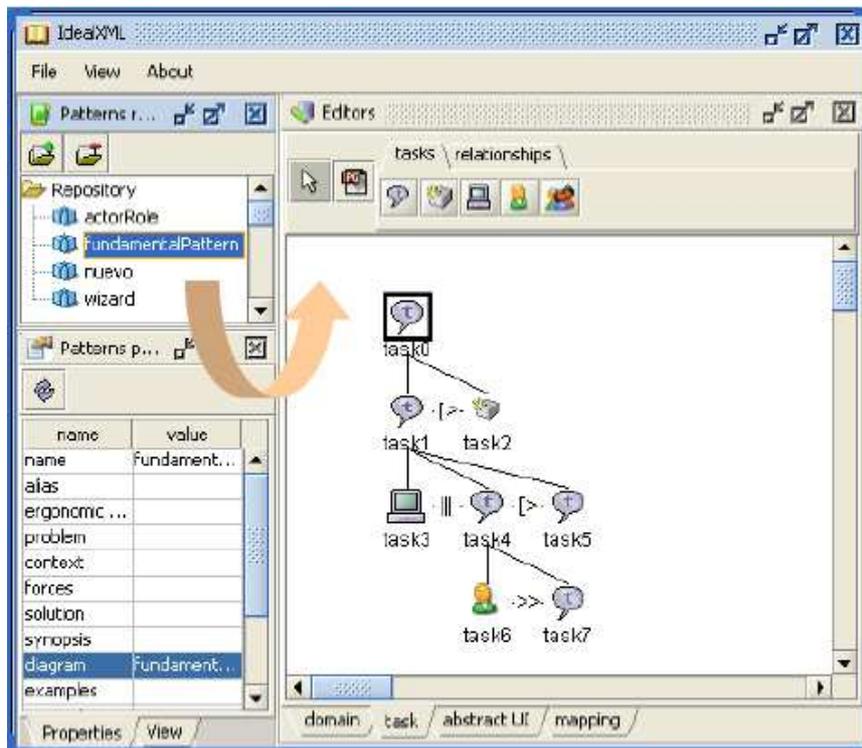


Figura 3-2. Editor de modelo de tareas incluido en IDEALXML

La herramienta construida, presentada con el nombre de *CoCTT*, se ha basado en el editor de tareas de la aplicación *IdealXML* (Montero 2005). Esta ha servido de base para la construcción de la herramienta colaborativa que se presenta en este Proyecto Final de Carrera.

3.4. coCTT

En Ingeniería del Software se dispone de un lenguaje unificado de modelado (UML) pero en Interacción Persona Ordenador no hay algo parecido ya que cada autor propone una solución; Extender el uso de UML, lenguajes de descripción de interfaces de usuario o notaciones propias como CTT (Paternó, 1999) que da soporte a la labor del modelado de tareas. Estas últimas ya están soportadas por algunas herramientas (CTTE ó IdealXML) pero sin duda queda mucho camino por recorrer pues la forma de trabajo está cambiando a pasos agigantados y ya no se requiere la presencia física del equipo de trabajo en un lugar para llevar a cabo estas tareas. Teniendo en cuenta que lo que predomina hoy en día es el trabajo en grupo, la herramienta propuesta en este Proyecto pretende fijar metas adicionales para permitir esta nueva forma de trabajo y permitir un seguimiento del proceso

de construcción de trabajo común eficaz. Por ello, se propone la mejora de, en concreto, la herramienta IdealXML para hacer de ella una herramienta colaborativa en el ámbito de la especificación y el modelado de tareas. En el Capítulo 4 se presenta de forma más detallada el proceso llevado a cabo para la construcción de esta herramienta.

Capítulo 4

Detalles de implementación

4.1. INTRODUCCIÓN

A la hora de formar un grupo de trabajo, no sólo en el ámbito del modelado de tareas, es necesario un alto grado de coordinación, que solamente es posible mediante una interacción completa de los miembros del equipo, en tiempo y espacio, para realizar una labor eficiente. Muchas veces esta coordinación no es posible, y esta circunstancia aumenta a medida que se instaura la filosofía del trabajo a distancia. Es por esto que, sin dejar de lado las ventajas que esta nueva forma de trabajar supone, la herramienta a desarrollar debe proporcionar, además de un entorno de trabajo específico, un entorno cooperativo y/o colaborativo común en el que los participantes mantienen su distancia física, pero son capaces de interactuar y coordinar sus movimientos.

En la construcción de la herramienta se deben de abordar además de los requisitos necesarios para la construcción de modelos de tareas, otros requisitos adicionales que permiten la colaboración entre usuarios pertenecientes a un equipo de trabajo.

La forma en la que se ha realizado el diseño para construir la aplicación final está inspirada en la Ingeniería del Software y ha seguido una serie de etapas contempladas en los procesos habituales de desarrollo software (Figura 4-1).

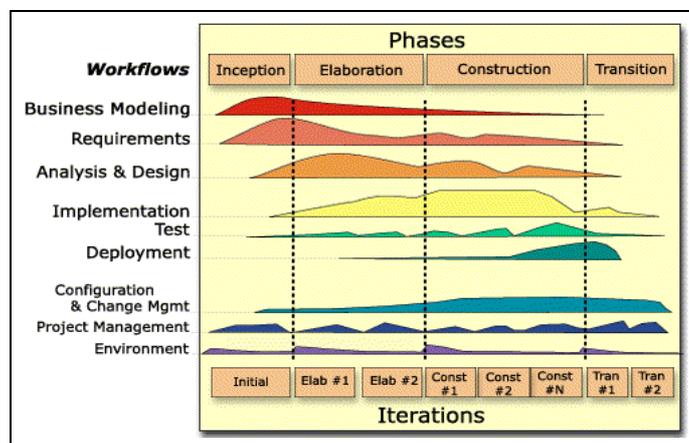


Figura 4-1. Desarrollo de software siguiendo el Proceso Unificado de Rational (Jacobson et al., 1999; Krutchen et al., 2000)

El Proceso Unificado utiliza UML para expresar gráficamente todos los esquemas de un sistema software. Los aspectos que definen un proceso unificado son tres:

- i. Dirigido por casos de uso:** El modelo de casos de uso describe la funcionalidad completa del sistema. Este modelo reemplaza la especificación funcional tradicional del sistema, es decir, "qué debe hacer el sistema", pero los casos de uso no son sólo una herramienta para especificar los requisitos del sistema, sino que también guían su diseño, implementación y testado, en definitiva guían el proceso software. Basado en el modelo de casos de uso, los desarrolladores crean modelos de diseño e implementación que realizan los casos de uso, los desarrolladores revisan cada uno de los modelos creados para comprobar si se corresponden con el modelo de casos de uso especificado. En este sentido los casos de uso no solo inician el proceso software sino que permanecen ligados a él durante todo el ciclo de vida.

- ii. Centrado en la arquitectura:** Los casos de uso por si solos no son suficientes. Se necesita algo más para llevar a un sistema que funcione y es aquí donde interviene la arquitectura. Podemos pensar en ella como la visión común que todos los participantes en el proyecto deben compartir y aceptar. La arquitectura nos da una perspectiva clara del sistema global, la cual es necesaria para controlar su desarrollo.

- iii. Iterativo e incremental:** Dado que los proyectos software son largos, es habitual dividir el trabajo en miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, y los incrementos a un crecimiento en el producto. Para ser más efectivas las iteraciones deben ser controladas, es decir, deben ser seleccionadas y llevadas a cabo de una forma planeada, de forma que cada una de las iteraciones constituye un miniproyecto software.

4.2. SITUACIÓN ACTUAL

Son muchos los programas necesarios a la hora de desarrollar una aplicación software; "Requisite Pro" en los requisitos, "Rational Rose" en el diseño o "Rational Robot" para las pruebas son solo algunos de los programas utilizados en Ingeniería

del software pero casi todos tienen una característica común, y es que son aplicaciones monousuario. En Interacción Persona-Ordenador (IPO) se proponen otras herramientas ya comentadas anteriormente, como CTTE (Mori et al,2004) ó IdealXML (Montero 2005) entre otras muchas y concretamente relacionadas con el modelado de tareas, pero al igual que las anteriores son herramientas pensadas para el uso exclusivo de una única persona. Puesto que para la obtención de una aplicación software, desde la recogida de requisitos hasta la implementación y pruebas finales, es necesaria la participación de múltiples trabajadores, sería útil pensar en herramientas colaborativas que faciliten esta labor.

4.3. ALCANCE Y JUSTIFICACIÓN

En este Proyecto Final de Carrera se pretende dar un pequeño paso adelante a la hora de llevar a cabo una especificación y modelado de tareas con notación CTT (Paternó 1999), realizando una aplicación que permita la interacción y colaboración entre los usuarios a la hora de llevarlo a cabo. Se ha realizado una herramienta colaborativa, en la que los miembros del equipo puedan interactuar entre sí en el mismo dominio de tiempo pero en diferente lugar intercambiando ideas y realizando modelos comunes. Para realizar esta labor, se ha construido una herramienta basada en la arquitectura cliente-servidor con múltiples clientes interaccionando con el servidor, el cual, se ejecutará de forma permanente y servirá la información a todos los clientes en ejecución.

Una herramienta colaborativa puede ser positiva en el sentido de que los miembros del equipo pueden exponer sus ideas al resto, compartir el trabajo ya realizado ó ayudar a otros resolviendo posibles dudas para dotar de una mayor calidad al trabajo final. La herramienta también pretende acortar distancias entre los miembros de un equipo de manera que estos se sientan integrados y útiles a la hora de realizar el trabajo, aumentando la confianza y sintiéndose respaldados en todo momento a la hora de afrontar sus quehaceres en el ámbito de la construcción de modelos de tareas con notación CTT (Paternó 1999).

4.4. OBJETIVOS

Se han marcado una serie de objetivos que a grandes rasgos marcarán el camino para conseguir una solución satisfactoria.

OBJ-0001: Inicio de Sesión: Puesto que serán varios los trabajadores que lleven a cabo la especificación y el modelado de tareas, estos deben de iniciar sesión en el sistema con un nombre distinto al del resto para poder ser identificados por los demás usuarios. Así mismo este hecho deberá de ser comunicado al resto de usuarios que se encuentren en ese momento registrados en el sistema.

OBJ-0002: Comunicación entre usuarios: La herramienta debe permitir una comunicación síncrona entre usuarios. Estos podrán comunicarse de una forma precisa y sencilla, pese a no estar en un mismo lugar.

OBJ-0003: Editor CTT colaborativo: El fin de la herramienta no es otro, que el desarrollar una especificación y modelado de tareas, por lo que este debe ser el objetivo principal de la aplicación. Se utilizará la notación CTT para llevar a cabo las tareas de modelado, manteniendo todas las opciones de las herramientas existentes y añadiéndole nueva funcionalidad, para que estas tareas puedan realizarse de forma colaborativa por varios miembros de un equipo de trabajo.

OBJ-0004: Incorporación tardía de usuarios: Dado que es posible que algunos trabajadores se incorporen una vez iniciado el trabajo, la herramienta debe permitir recuperar el estado de las tareas de modelado, para que los usuarios puedan incorporarse en la menor brevedad posible y continuar con sus labores.

OBJ-0005: Abrir/Guardar documentos: La herramienta permitirá salvar los trabajos desarrollados por los usuarios así como volverlos a cargar una vez se quiera reanudar trabajos anteriores.

OBJ-0006: Visualización XML de los documentos: Los documentos relativos al modelado de tareas se guardarán en formato XML, por ello, se debe de contemplar la opción de poder obtener una visualización del documento en este formato mientras se está realizando las tareas de modelado..

OBJ-0007: Personalización del editor de Chat: El color de letra de los mensajes así como la intensidad de esta podrá ser elegido por cada usuario, así como incluir iconos que muestren su estado: uno que indique que ese usuario está al teléfono y otro que muestre que está manteniendo una conversación privada.

OBJ-0008: Finalización de Sesión: Una vez se haya concluido el trabajo, la herramienta permite cerrar la sesión del usuario comunicándole al resto esta situación.

4.5. ESPECIFICACIÓN DE REQUISITOS SOFTWARE

A continuación se detalla la especificación de requisitos software recogidos para llevar a cabo el posterior diseño e implementación de la herramienta:

IRQ-0001: Entrada en el sistema (dependencia de OBJ-0001: Inicio de Sesión). Cuando un usuario desee conectarse al sistema, este le pedirá un nombre de usuario para conectarse, a continuación el sistema comprobará si en ese momento hay otro usuario utilizando ese mismo nombre, si no es así el usuario quedará registrado y se advertirá, mediante un mensaje escrito en el propio Chat al resto de usuarios conectados en ese momento de la nueva entrada al sistema. Si por el contrario el nombre estuviese en uso, se le pediría al usuario que introdujese un segundo nombre. En la Tabla 4-1 se muestran los datos específicos para la realización del requisito IRQ-0001.

Datos específicos
Tipo de mensaje enviado
Nombre de usuario

Tabla 4-1. Datos específicos en la entrada al sistema

IRQ-0002: Comunicación (dependencia de OBJ-0002: Comunicación entre usuarios y OBJ-0007: Personalización del editor de Chat). Se debe de disponer de una facilidad tipo "Chat" para que los usuarios puedan comunicarse entre ellos. Cada mensaje que envíe el usuario, irá precedido del nombre de dicho usuario para distinguirlo del resto de mensajes y se actualizarán el editor de mensajes de cada usuario con el nuevo mensaje enviado. También se debe de contemplar la

posibilidad del envío de mensajes privados, es decir, mensajes que recibirá exclusivamente una única persona.

En esta parte, también debe haber un apartado destinado a la visualización de usuarios conectados en ese momento, así como opciones que permitan la personalización del Chat; una paleta de colores donde los usuarios puedan personalizar el color del texto escrito a su gusto, así como la intensidad del mensaje enviado (negrita). En la Tabla 4-2 se muestran los datos específicos relacionados con el requisito IRQ-0002.

También debe distribuirse al resto de usuarios conectados la información relativa al modelo de tareas.

Datos específicos
Tipo de mensaje enviado
Nombre de usuario que envía el mensaje
Receptor del mensaje (todos/privado)
Mensaje de texto
Color texto
Negrita (Si/No)
Modelo de tareas

Tabla 4-2. Datos específicos en la comunicación entre usuarios

IRQ-0003: Modelado de tareas CTT (dependencia OBJ-0003: Editor CTT colaborativo). La herramienta permite realizar modelos de tareas basándose en la notación CTT (Paternó 1999). Los iconos utilizados en el modelo de tareas son similares a los utilizados en la herramienta CTTE.

Tanto las tareas como las dependencias entre ellas (Tabla 4-3), podrán manipularse debiendo quedar este hecho reflejado en los editores del resto de participantes. También disponen de unas propiedades que pueden ser actualizadas en todo momento. Todos los cambios y modificaciones que se realicen deben de ser coherentes en los editores de cada usuario.

Icono	Descripción
	Iconos con los que se pueden definir los distintos tipos de tareas (abstracta, aplicación, interacción, cooperación y usuario)
[] >> []>> [] [> > =	Iconos asociados a las dependencias entre tareas (conurrencia independiente, sincronización, activación, activación con paso de información, elección, desactivación, suspensión/reinicio, independencia de orden)
[T] T*	Iconos asociados a las tareas unarias que pueden definirse utilizando el editor de modelos de tareas: Tarea opcional e Iteración.

Tabla 4-3. Iconos asociados a las tareas y operadores temporales

La información que habrá que pasar al servidor para que este la envíe al resto de clientes, dependerá del tipo de acción realizada (Tabla 4-4):

Datos específicos		
Crear Tarea Editar tarea	Posición	
	Nombre	
	Tipo	
	Propiedades	frequency
		importance
		structurationLevel
		userAction
		taskItem
		criticity
		centrality
optional		
Iterative		
comlesityLevel		
Seleccionar tarea	Numero relativo a la tarea seleccionada	
Borrar tarea	Numero relativo a la tarea seleccionada	
Mover tarea	Posición	
Crear Relación Editar Relación	Propiedades	id
		nombre
		source
		target
		precondition
		passedObject
Seleccionar Relación	Numero relativo a la relación seleccionada	
Tipo de mensaje enviado		

Tabla 4-4. Acciones y datos asociados al modelado de tareas

IRQ-0004: Incorporación de usuarios tardío: (dependencia OBJ-0004: Recuperar estado actual). Se puede dar el caso, de que un usuario se conecte una vez haya comenzado el trabajo. Si esto ocurre, en el editor de tareas del usuario aparecerá el diagrama que se haya elaborado en ese momento para que el usuario

pueda continuar el trabajo con el resto de sus compañeros y será el primer usuario conectado al sistema, el encargado de enviarle toda la información necesaria al compañero recién incorporado. Los datos relativos al IRQ-0004 se muestran en la Tabla 4-5.

Datos específicos
Tipo de mensaje enviado
Datos específicos de modelado de tareas (Tabla 4)
Usuario receptor

Tabla 4-5. Datos específicos en al incorporación de tareas tardías

IRQ-0005: Gestión de documentos (dependencia OBJ-0005: Abrir/Salvar documentos y OBJ-0006: Visualización XML de los documentos). Los modelos de tareas que los usuarios realicen se podrán guardar/visualizar en formato xml así como abrirlos para su posterior reutilización. Una vez abierto un diagrama se abrirá en el resto de editores clientes. Se utilizarán los datos mostrados en la Tabla 4-6 para la realización del IRQ-0006.

Datos específicos
Tipo de mensaje enviado
Usuario que envía petición de abrir

Tabla 4-6. Datos específicos en la gestión de Documentos

IRQ-0006: Salida del sistema (dependencia OBJ-0008: Finalización de Sesión). Cuando el usuario salga del sistema, se debe de comunicar este hecho al resto de participantes mediante un mensaje en su editor de Chat. El nombre utilizado por el usuario quedará libre para que otro usuario, o ese mismo en otro momento, lo pueda utilizar. Se utilizarán los datos mostrados en la Tabla 4-7 para la realización del IRQ-0006.

Datos específicos
Tipo de mensaje enviado
Nombre usuario

Tabla 4-7. Datos específicos a la salida del sistema

4.6. VISTA DE CASOS DE USO

En esta vista se presenta el comportamiento del sistema y se encuentran los modelos relacionados con la captura de requisitos. Un caso de uso es un escenario de utilización del sistema por parte de un actor externo (usuario del sistema). La definición de caso de uso (Jacobson et al, 1999) incluye:

- Un comportamiento del sistema
- Una secuencia de transacciones relacionadas entre el actor y el sistema
- Proporcionar un servicio relevante al actor

Los Casos de uso son una herramienta útil en la captura de requerimientos, comunicación con los expertos del dominio del problema y las pruebas del sistema.

4.6.1. Actores

Representa a los usuarios que interactúan con la aplicación. En este caso, el actor usuario (Figura 4-2), representa a cualquier ingeniero de software destinado a utilizar la herramienta.



Figura 4-2. Icono asociado a un actor/usuario

4.6.2. Diagramas de casos de uso

Diagrama de Casos de Uso del subsistema Cliente: El subsistema cliente es el encargado de llevar a cabo las siguientes tareas.

- Entrada al sistema
- Comunicación entre usuarios
- Modelado de tareas CTT
- Gestión de documentos
- Personalizar Chat
- Salida del sistema

Para llevar a cabo estas acciones se ha realizado un diagrama de casos de uso relativo al subsistema cliente (Figura 4-3).

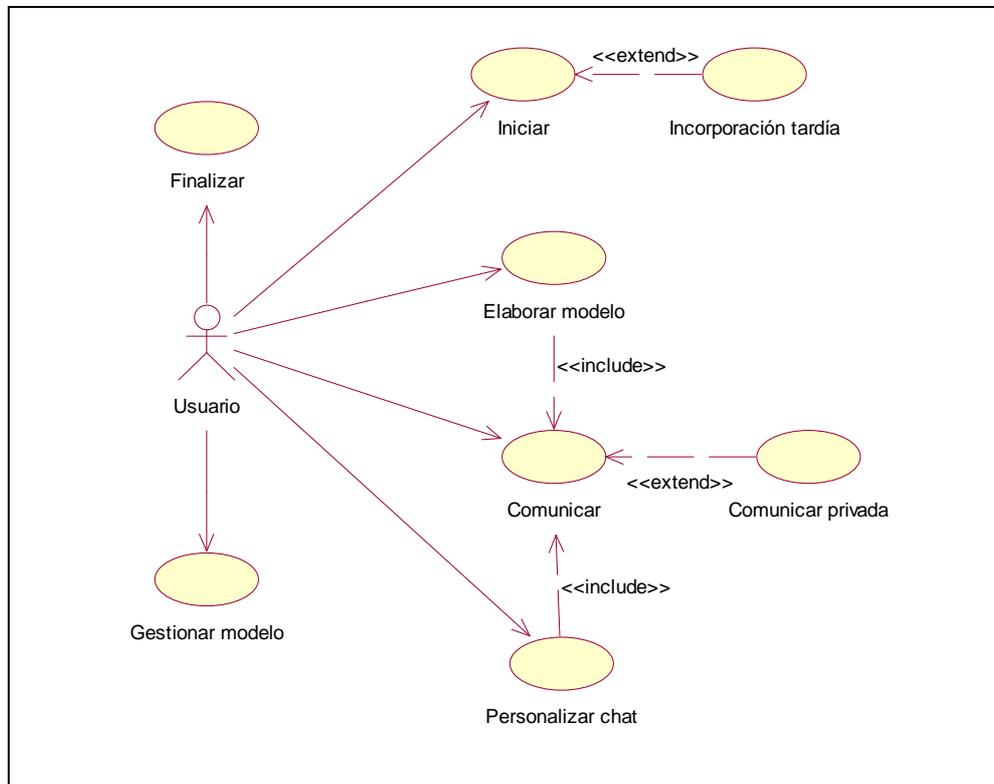


Figura 4-3. Diagrama de casos de uso del subsistema cliente

La descripción referente a cada caso de uso queda definida en las Tablas 4-8 a 4-14:

UC-0001	Iniciar	
Dependencias	[OBJ-0001] Iniciar Sesión	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario desee trabajar en el sistema	
Precondición	No puede haber otro usuario en el sistema con el mismo nombre	
Secuencia normal	Paso	Acción
	1	El sistema pide el nombre con el que el usuario se quiere conectar
	2	El actor usuario introduce el nombre
	3	El sistema comprueba si hay más usuarios conectados en el sistema
	4	El actor usuario queda registrado en el sistema
Poscondición	Ninguna	
Excepciones	Paso	Acción
	2	Si el nombre aportado por el usuario ya existe en el sistema este pedirá al usuario que introduzca otro nombre distinto.
	3	Si existen más usuarios conectados en el sistema se realiza el caso de uso UC-0004 <i>Incorporación tardía</i> .

Tabla 4-8. Caso de uso UC-0001: iniciar

UC-0002		Comunicar	
Dependencias	[OBJ-0002] Comunicación síncrona (Chat) [OBJ-0003] Editor CTT colaborativo [OBJ-0007] Personalización del editor de Chat		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se quieran comunicar acciones al resto de usuarios.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El cliente envía al servidor información relativa a: <ul style="list-style-type: none"> • Un mensaje o un símbolo • Diagrama de modelado (Se realiza el <i>UC-0003 Elaborar modelo ó UC-Gestionar Modelo</i>) • Personalizar Chat (Se realiza el <i>UC-0006 Personalizar Chat</i>) 	
	2	El servidor envía dicha información al resto de aplicaciones clientes	
	3	La información es actualizada	
Poscondición	Ninguna		
Excepciones	Paso	Acción	

Tabla 4-9. Caso de uso UC-0002: comunicar

UC-0003		Elaborar modelo	
Dependencias	[OBJ-0003] Editor CTT colaborativo [OBJ-0002] Comunicación entre usuarios		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario deseé trabajar con el editor de tareas		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	
	1	El usuario realiza cualquiera de las opciones posibles descritas en el requisito <i>IRQ-0003 Modelado de tareas CTT</i>	
	2	Se realiza el caso de uso <i>UC-0002 Comunicar</i>	
Poscondición	Ninguna		
Excepciones	Paso	Acción	

Tabla 4-10. Caso de uso UC-0003: elaborar modelo

UC-0004		Incorporación tardía	
Dependencias	[OBJ-0001] Iniciar [OBJ-0004] Incorporación de usuarios tardíos		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario se conecte habiendo otros usuarios ya conectados en el sistema		
Precondición	Debe haber otros usuarios conectados en el sistema		
Secuencia normal	Paso	Acción	
	1	El sistema recupera la información del diagrama del modelado de tareas del primer usuario conectado	
	2	El editor de tareas del usuario recién incorporado es actualizado	
Poscondición	Ninguna		
Excepciones	Paso	Acción	

Tabla 4-11. Caso de uso UC-0004: incorporación tardía

UC-0005	Gestionar modelo	
Dependencias	[OBJ-0005] Abrir/Guardar documentos [OBJ-0006] Visualización XML de los documentos	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee guardar o abrir documentos relativos a los diagramas de modelado de tareas así como la visualización XML de estos.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El usuario indica al sistema si desea abrir, guardar o visualizar el documento en formato XML.
	2	El sistema le devuelve su petición
Poscondición	Ninguna	
Excepciones	Paso	Acción
	2	Si la opción es abrir, se realiza el caso de uso <i>UC-0002 Comunicar</i> y el fichero es abierto en el resto de sesiones.

Tabla 4-12. Caso de uso UC-0005: Gestión del modelo

UC-0006	Personalizar Chat	
Dependencias	[OBJ-0002] Comunicar [OBJ-0007] Personalización del editor de Chat	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario modifique el aspecto de su editor de Chat.	
Precondición	Ninguna	
Secuencia normal	Paso	Acción
	1	El usuario indica al sistema el color del texto con el que desea comunicarse o si quiere enviar texto destacado (negrita)
	2	Se realiza el caso de uso <i>UC-0002 Comunicar</i> .
	3	El color del texto del usuario queda actualizado.
Poscondición	Ninguna	
Excepciones	Paso	Acción

Tabla 4-13. Caso de uso UC-0006: Personalizar char

UC-0007	Finalizar	
Dependencias	[OBJ-0007] Finalizar sesión	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee finalizar su sesión en el sistema	
Precondición	El usuario debe de estar previamente registrado en el sistema	
Secuencia normal	Paso	Acción
	1	El usuario indica al sistema que desea salir
	2	El sistema finaliza la sesión del usuario.
Poscondición	Ninguna	
Excepciones	Paso	Acción

Tabla 4-14. Caso de uso UC-0007: Finalizar

Diagrama de Casos de Uso del subsistema Servidor: El servidor debe de estar en estado "conectado" para poder establecer las correspondientes comunicaciones. El diagrama correspondiente a este caso de uso se muestra en la Figura 4-4y su principal función será la de ocuparse de gestionar los mensajes enviados por los

usuarios para hacérselos llegar al resto. Su funcionamiento queda descrito de una forma más detallada en los diagramas de secuencia que se muestran más adelante.

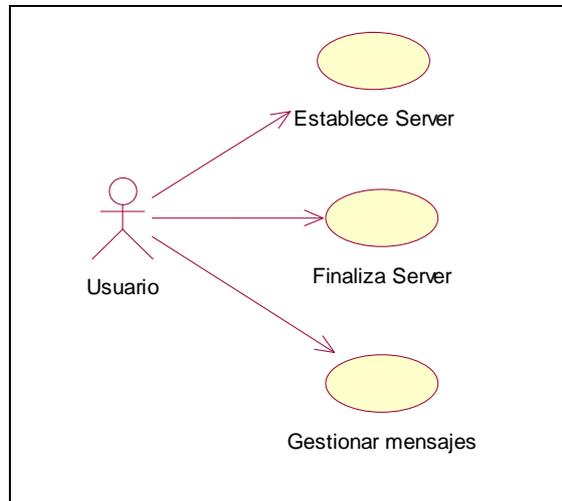


Figura 4-4. Diagrama de casos de uso del subsistema servidor

4.7. MATRIZ DE RASTREABILIDAD OBJETIVOS / REQUISITOS

Este apartado contiene las matrices *requisitos-elementos de modelado*, conocidas como matrices de rastreabilidad, de forma que para cada requisito identificado en el *Documento de Requisitos del Sistema* se pueda conocer con qué tipos, asociaciones y operaciones del sistema está asociado. El formato de las matrices de rastreabilidad puede verse en la Tabla 4-15 y la Tabla 4-16.

TRM-0001	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006	OBJ-0007
IRQ-0001	↑	-	-	-	-	-	-
IRQ-0002	-	↑	-	-	-	-	↑
IRQ-0003	-	-	↑	-	-	-	-
IRQ-0004	-	-	-	↑	-	-	-
IRQ-0005	-	-	-	-	↑	↑	-
IRQ-0006	-	-	-	-	-	↑	-

Tabla 4-15. Matriz de rastreabilidad. Objetivos y requisitos de información

TRM-0002	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006	OBJ-0007
UC-0001	↑		-	-	-	-	-
UC-0002	-	↑	↑	-	-	-	↑
UC-0003	-	↑	↑	-	-	-	-
UC-0004	↑	-	-	↑	-	-	
UC-0005	-	-	-	-	↑	↑	-
UC-0006	-	↑	-	-	-	↑	-
UC-0007	-	-	-	-	-	-	↑

Tabla 4-16. Matriz de rastreabilidad: Objetivos y requisitos funcionales

4.8. DIAGRAMAS DE SECUENCIA

A continuación se muestran algunos diagramas de secuencia para expresar de forma más clara cómo se realiza la comunicación entre el cliente y el servidor, ya que mientras que los diagramas de casos de uso permiten el modelado de una vista "business" del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes pasados entre los objetos. El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema.

4.8.1. Inicio de sesión de un usuario en el sistema

El funcionamiento relativo a este caso se encuentra reflejado en el diagrama de secuencia que muestra la Figura 4-5. Antes de poder recibir y enviar mensajes, el cliente de Chat debe "conectar" con el servidor. El servidor esta siempre a la escucha de posibles peticiones por parte de los clientes de Chat. El cliente, enviará la petición de conectar por un puerto junto con el nombre que el usuario desee utilizar en el sistema. El servidor, que tendrá una lista de los nombres que ya están siendo usados, averiguará si el nombre proporcionado es válido (no coincide con ningún nombre usado) o no. Si fuese válido, se crea un nuevo hilo de ejecución (HiloServidor1) que se ocupa de enviar y recibir mensajes de ese cliente, así como informar al resto de hilos (en este caso HiloServidor2) ya creados de la entrada de

un nuevo usuario en el sistema. En caso contrario, el servidor advierte al cliente de que el nombre no es válido y éste deberá introducir otro nombre en el sistema. Una vez realizada la operación, el servidor vuelve a quedar a la escucha de peticiones de nuevos clientes por el puerto.

El hilo servidor recién creado (HiloServidor1), se encarga de recibir ahora los mensajes y pasarlos al Servidor para que este los reenvíe a los HiloServidor pertinentes.

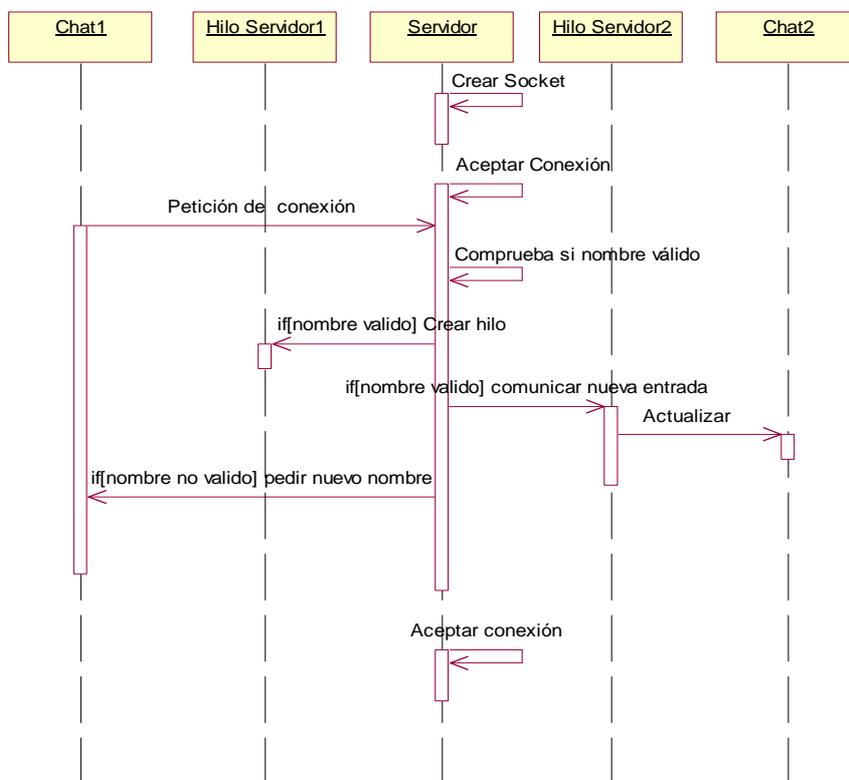


Figura 4-5. Diagrama de secuencia correspondiente al inicio de sesión

Cada vez que un nuevo usuario se conecte en el sistema se creará una nueva sesión y su hilo correspondiente que se comunicará con el servidor.

4.8.2. Comunicar

El diagrama de secuencia de la actividad correspondiente a comunicar se muestra en la Figura 4-6 y su funcionamiento se describe como sigue: El cliente que quiera comunicarse con el resto enviará el mensaje correspondiente a su HiloServidor1, que lo entrega al Servidor para que lo reenvíe al resto de clientes. El Servidor reenvía el mensaje a todos los HiloServidor, que los entregan a sus respectivos clientes-Chat. En el caso del ejemplo, el cliente Chat2 recibe el mensaje que ha sido enviado por el cliente Chat1.

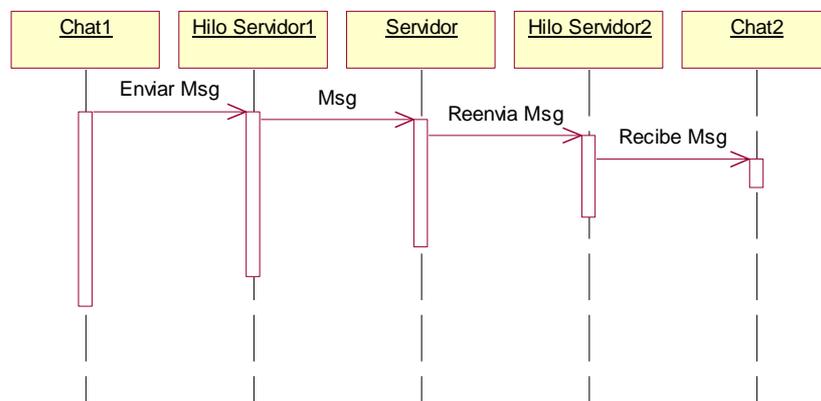


Figura 4-6. Diagrama de secuencia asociado al proceso de comunicación

Se debe tener en cuenta que si la comunicación es privada, el cliente que envía el mensaje indicará el nombre del receptor, por lo que el servidor únicamente entregará el mensaje a este último.

4.8.3. Incorporación tardía

Al conectarse cada usuario, el servidor comprueba si ya hay más usuarios conectados en el sistema. Si es así, pide la información relativa al modelo de tareas al primer usuario conectado, pues este es el que dispondrá de toda la información y será el que le pueda enviar la información a través del servidor, al nuevo usuario conectado, (Figura 4-7).

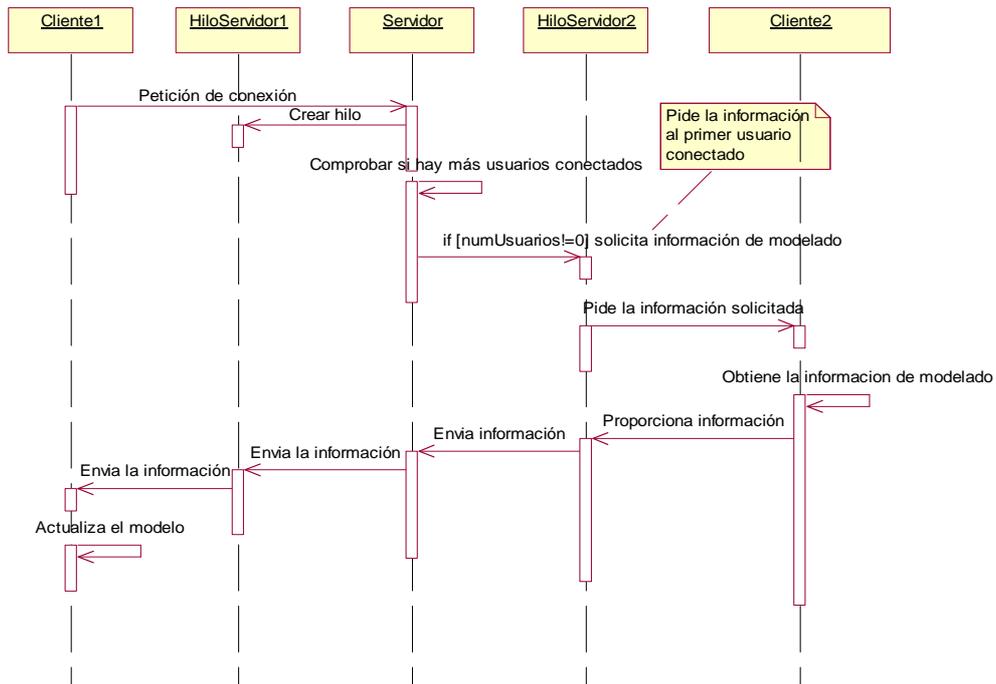


Figura 4-7. Diagrama de secuencia correspondiente a un escenario de incorporación tardía

4.8.4. Finalizar sesión

Cuando el usuario finaliza la sesión, el servidor se encargará de darle de baja en el sistema, así como comunicar al resto de usuarios este hecho, (Figura 4-8).

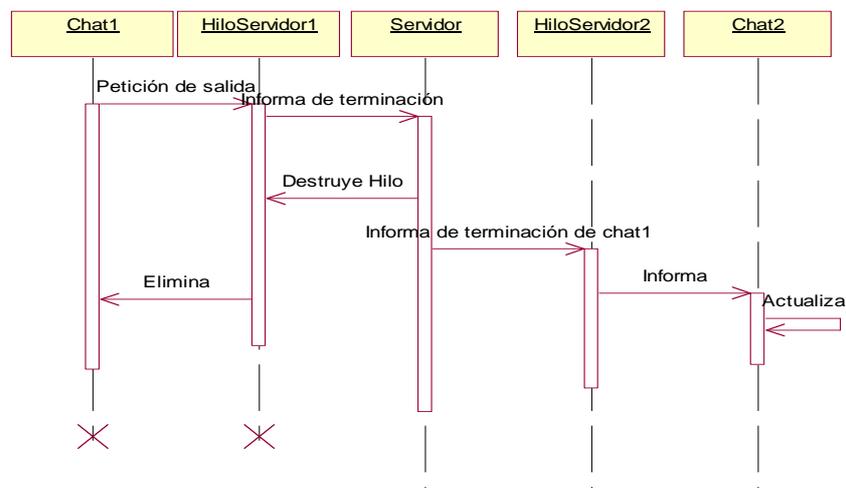


Figura 4-8. Diagrama de secuencia correspondiente al cierre de sesión

4.9. DISEÑO DEL SISTEMA

Para implementar la arquitectura cliente-servidor se han desarrollado los siguientes paquetes:

- **Subsistema Cliente:** El subsistema Cliente se ha construido utilizando los siguientes paquetes:
 - *EditorCTT*: Implementa la funcionalidad necesaria para la construcción de modelos de tareas.
 - *Mediador*: Actúa de intermediario en la comunicación entre el cliente y el paquete encargado de realizar el modelo de tareas con el cliente.
 - *Cliente*: El paquete cliente está formado por dos clases: La clase *cliente*; es la encargada de enviar toda la información al servidor para que este la distribuya al resto de usuarios y la clase *sesión*, que se comunica con el servidor para que este le proporcione la información necesaria y permita mantener al cliente asociado a la sesión actualizado.

La Figura 4-9 muestra el diagrama de clases donde se puede ver gráficamente las relaciones de *dependencia* y *asociación* que se establecen entre clases. Las primeras se utilizan cuando una clase se limita a utilizar un objeto de otra, pero si por el contrario, el conocimiento del objeto utilizado es más profundo y se utilizan métodos de este objeto, se establece en este caso una relación de asociación.

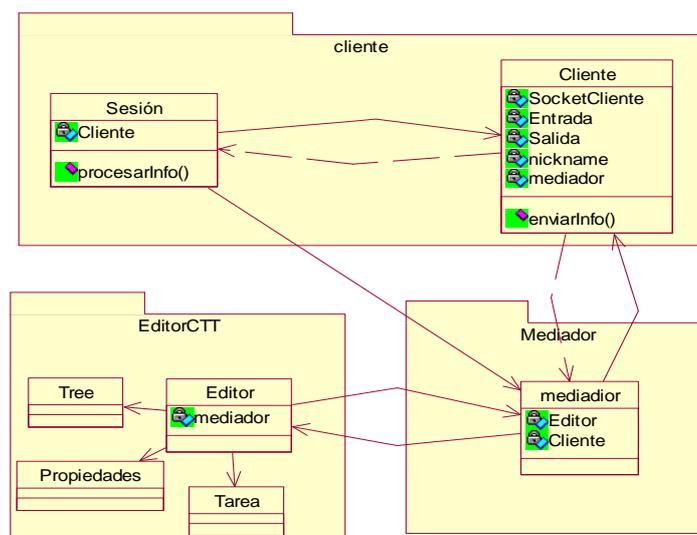


Figura 4-9. Diagrama de paquetes del subsistema cliente

En este caso, el sistema es multicliente, es decir, se ejecutan varios clientes a la vez (normalmente en varias máquinas) que interactúan con un mismo servidor.

- **Subsistema Servidor:** Al servidor le llegará información del cliente y será el encargado de procesar la información y distribuirla al resto de sesiones para que se mantengan actualizadas, (Figura 4-10).

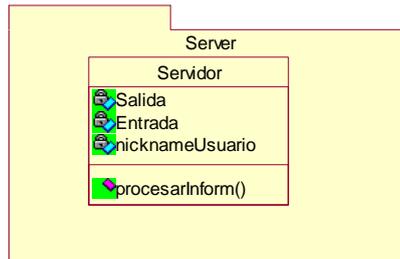


Figura 4-10. Diagrama de paquetes del subsistema servidor

En programación distribuida, un conjunto de ordenadores conectados por una red son usados colectivamente para realizar tareas distribuidas. La implementación de los sistemas distribuidos puede ser realizada usando dos modelos: El modelo cliente-servidor y el modelo basado en objetos.

El modelo cliente-servidor contiene un conjunto de procesos clientes y un conjunto de procesos servidor y se precisan además de unos recursos (software), que son manejados por los procesos servidor. Cliente y Servidor deben hablar el mismo lenguaje para conseguir una comunicación efectiva, el primero solicita al segundo unos recursos y este último los concede, le hace esperar o lo deniega según los permisos que tenga.

En el modelo orientado a objetos hay una serie de objetos que solicitan servicios (clientes) a los proveedores de los servicios (servidores) a través de una interfaz de encapsulación definida. Un cliente envía un mensaje a un objeto (servidor) y este decide que ejecutar, RMI y CORBA son algunos de estos sistemas basados en objetos.

La herramienta construida en este proyecto ha sido realizada utilizando el modelo cliente-servidor, siendo el elemento clave de la comunicación los sockets. El modelo orientado a objetos se expondrá como trabajo futuro en el Capítulo 5.

Los sockets son puntos finales de comunicación entre procesos. Los procesos los tratan como descriptores de ficheros, de forma que se pueden intercambiar datos con otros procesos recibiendo y transmitiendo a través de sockets. Una de las ventajas principales de la programación con sockets es su sencillez.

La implementación ha sido realizada en lenguaje Java (JDK 1.4.1) debido a que es una herramienta ideal para la programación distribuida gracias a su portabilidad, seguridad y amplio abanico de componentes. El entorno de desarrollo utilizado ha sido JBuilder 9 Enterprise.

Capítulo 5

Caso de estudio

En este Capítulo se proponen una serie de ejemplos que permitirán probar el funcionamiento de la herramienta coCTT, así como obtener unas conclusiones finales a cerca del funcionamiento de esta.

5.1. CoCTT: EJEMPLO 1. EJEMPLO PRÁCTICO

Varios usuarios pretenden realizar, de forma conjunta, un diagrama sobre las tareas a realizar en el suministro de productos de una máquina de café. Para resolver el problema hay tres usuarios conectados: Blanca, Juan y Carlos y uno de ellos se encarga de exponer el problema al resto, (Figura 5-1). El problema al que se enfrentan los usuarios ocupados de las tareas de análisis es el siguiente:

- La máquina de café está situada en lugares públicos para suministrar productos calientes a los ciudadanos. El usuario se acerca a la máquina en la que se muestran el tipo de producto y su precio. Posee una lista de precios, la ranura para las monedas y el selector de producto.
- En primer lugar se debe seleccionar el producto a tomar (café sólo, con leche, descafeinado o chocolate). Si hay existencias, se mostrará la cantidad de monedas a introducir. Cuando se introduzca la cantidad exacta, se pedirá si se quiere con azúcar o no. Una vez que se responde, se vierte el contenido seleccionado en una taza listo para tomar.
- Si el usuario no dispone de suficiente cantidad al introducir monedas, puede pulsar el botón de cancelar.

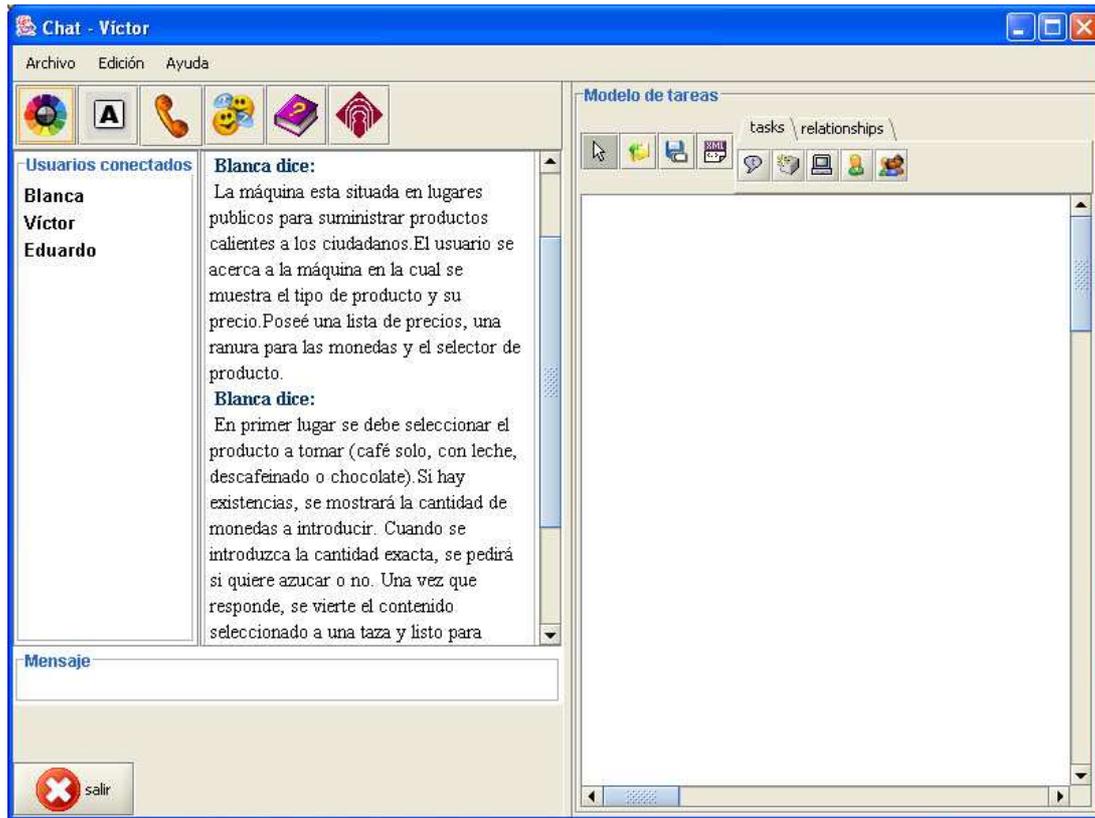


Figura 5-1. Planteamiento del problema mediante la aplicación CoCTT

Anteriormente se he mencionado en qué consiste el análisis de tareas. Este método se define como el estudio de lo que se requiere del usuario en términos de acciones y/o procesos cognitivos para completar una tarea. Así, un análisis detallado de tareas permitirá entender el sistema actual y los flujos de información en el mismo.

Los análisis de tareas hacen posible diseñar y organizar las tareas apropiadamente dentro del nuevo sistema. Las funciones a incluir dentro del sistema y de la interfaz de usuario pueden ser entonces especificadas con precisión.

Si es preciso, se establecen sesiones de observación para disponer de una perspectiva apropiada y se acude a los usuarios cuantas veces sea preciso para aclarar todas las cuestiones necesarias.

Una vez explicado el enunciado, los tres desarrolladores aportan ideas sobre cómo se podría lleva a cabo la resolución del problema, conociendo las reglas básicas necesarias para construir modelos de tareas con la herramienta CoCTT:

- Una tarea es abstracta si se descompone en varias subtareas de distinta naturaleza.
- Los operadores relacionan dos tareas. Si una tarea es más compleja se debe descomponer en un árbol.
- Permite guardar el análisis de tareas en xml.
- Permite abrir un diagrama que previamente ha sido guardado.
- Permite borrar o cambiar de posición las tareas dibujadas.

El objetivo aquí es dividir las tareas de alto nivel en subtareas y operaciones constitutivas. Para dividir la tarea, se formulará la pregunta "cómo se realiza esta tarea". Si se identifica una tarea a un nivel inferior, será posible construir la estructura preguntando "por qué se hace esto".

La descomposición de tareas se puede llevar a cabo mediante las siguientes etapas:

- Se identifica la tarea a analizar a partir de la lista de tareas.
- Se descompone entre 4 y 8 subtareas. Estas subtareas deberían estar especificadas en términos de objetivos y, entre ellas, deberían cubrir el área de interés.
- Se dibujan las subtareas como un diagrama asegurando que está completo.

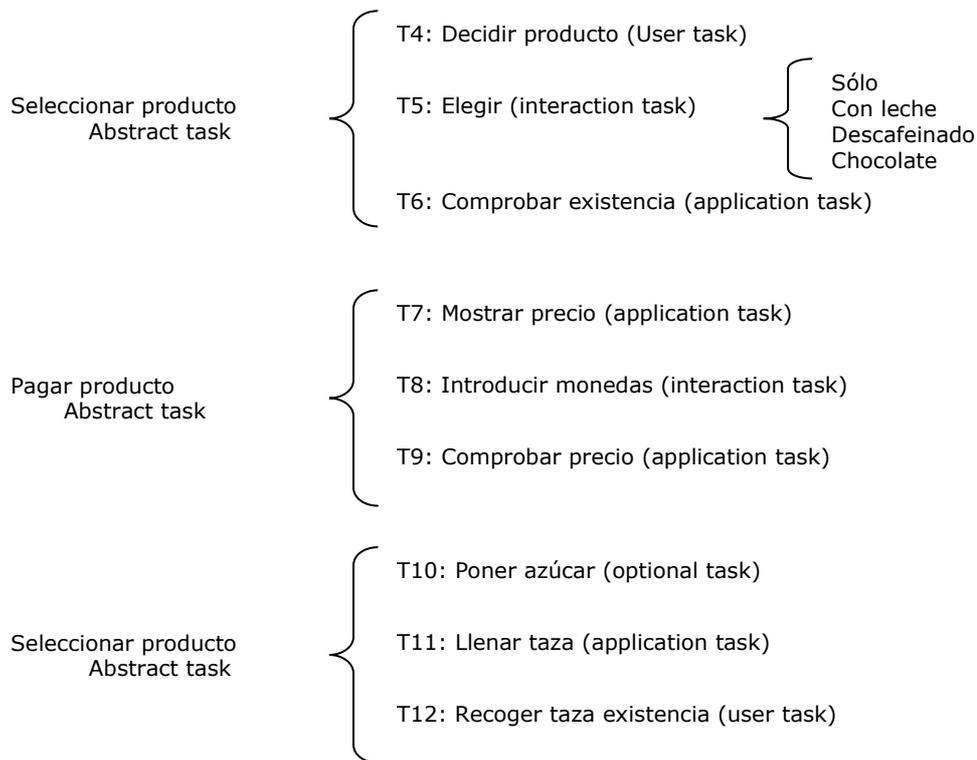
Hay que decidir sobre el nivel de detalle de la descomposición, lo que asegurará un tratamiento consistente de la situación. Podría decidirse que la descomposición de tareas continuará hasta que los flujos fueran representados con mayor facilidad en un diagrama de flujo de tareas.

Como se señaló previamente, es importante que el proceso de definición de requerimientos no se base en el simple registro de las operaciones existentes, sino que también aporte probables mejoras que resulten de la introducción de nuevos sistemas y facilidades.

Las tareas de alto nivel del ejemplo expuesto (Figura 5-2) podrían ser las siguientes:

Usuario {
T1: Seleccionar producto
T2: Pagar producto
T3: Servir producto

Cada usuario podría ocuparse de la descomposición en subtareas de las anteriores e intercambiar ideas con el resto sobre como quedaría el modelo final. El editor de Chat permite a los desarrolladores discutir sobre las tareas que serán llevadas a cabo para cubrir totalmente el enunciado. Los usuarios deciden que la división de tareas (Figura 5-2, Figura 5-3, Figura 5-4, Figura 5-5), se realizará del siguiente modo:



Todos los usuarios conectados han estado de acuerdo en realizar la anterior descomposición aunque han tenido que discutir en profundidad el tipo de relaciones de asociación que se establecerían entre ellas.

$T1 [] >> T2 [] >> T3$: El operador temporal es *Activación con paso de información* debido a que cuando termina $T1$ genera algún valor que se pasa a $T2$ antes de ser activada, de igual modo ocurre esto con $T3$. Las tareas café solo, con leche, descafeinado o chocolate se seleccionarán alternativamente y una vez que una esté activa las otras no estarán disponibles hasta que no termine la que está activa, por ello, se ha decidido poner el operador temporal *elección*.

$T4 >> T5 [] >> T6$: Entre $T4$ y $T5$ se establece la relación de activación, Cuando termina la $T4$ se activa la $T5$. Las dos tareas se realizan de forma secuencial y $T5$ no necesita que $T4$ le suministre ningún valor. En cambio, sí ocurre esto entre $T5$ y

T6 ya que el la elección del producto tiene que ser conocido por la tarea T6 para comprobar las existencias.

T7 []>> T8 []>T9: Entre T7 y T8 se utiliza el operador Activación con paso de información por el mismo motivo anterior. Entre T8 y T9 se utiliza el operador *desactivación* debido a que la primera tarea es desactivada cuando comienza la ejecución de la segunda, es decir, cuando la aplicación comprueba la cantidad introducida se deshabilita la opción introducir monedas. Se podría pensar también en una relación de activación es decir, cuando termina la tarea T1 se activa la tarea T2 y las dos tareas se realizan de forma secuencial, pero tras una breve charla entre los usuarios a cerca de cómo se llevarán a cabo las tareas, se decide por el operador *desactivación* debido a que la aplicación, calculará el precio siempre que el usuario introduzca una moneda para ver si este llega al límite exacto, para ello se tendrá que desactivar la opción introducir moneda, hasta que la aplicación calcule el precio. En este sentido, la herramienta coCTT ayuda a los desarrolladores a clarificar ideas en el momento del desarrollo y eliminar explicaciones posteriores que supondrían pérdida de tiempo.

[T11] []>> T12 []>> T13: Poner azúcar es una tarea opcional ya que el café se puede servir con azúcar o sin ella. Esta información debe ser pasada a la tarea llenar la taza, y una vez que esta esté llena se informará al usuario para que la recoja, de ahí los dos operadores temporales de activación con paso de información.

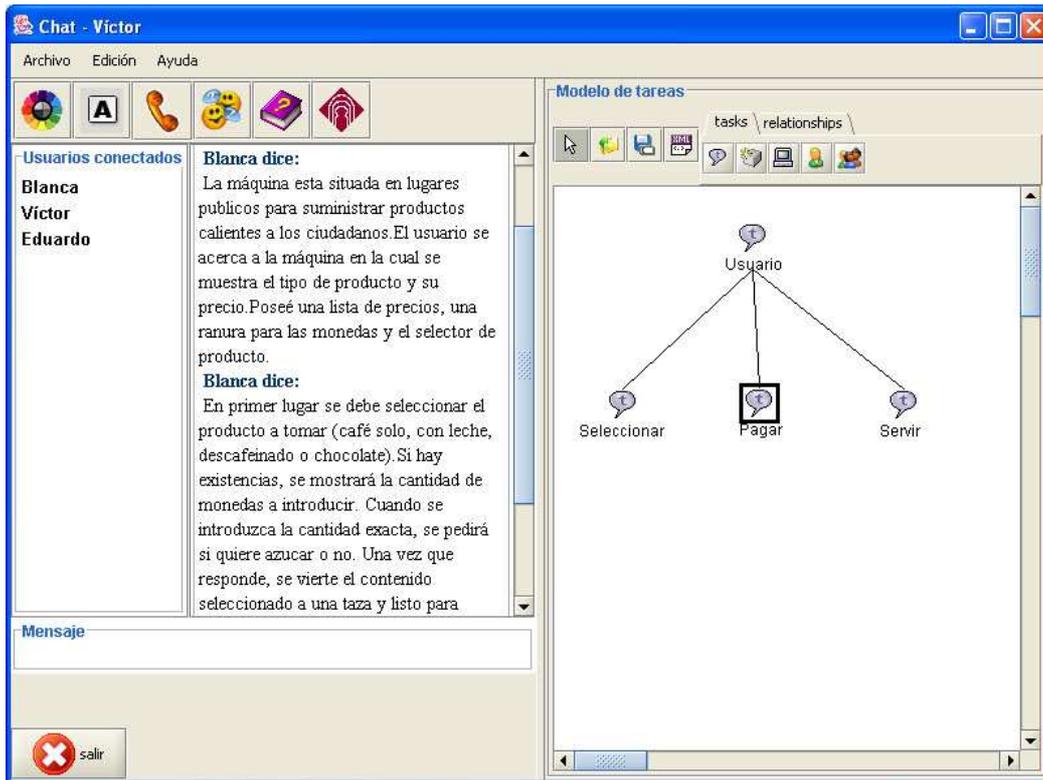


Figura 5-2. Resolución del problema con la herramienta CoCTT (Ejemplo 1)

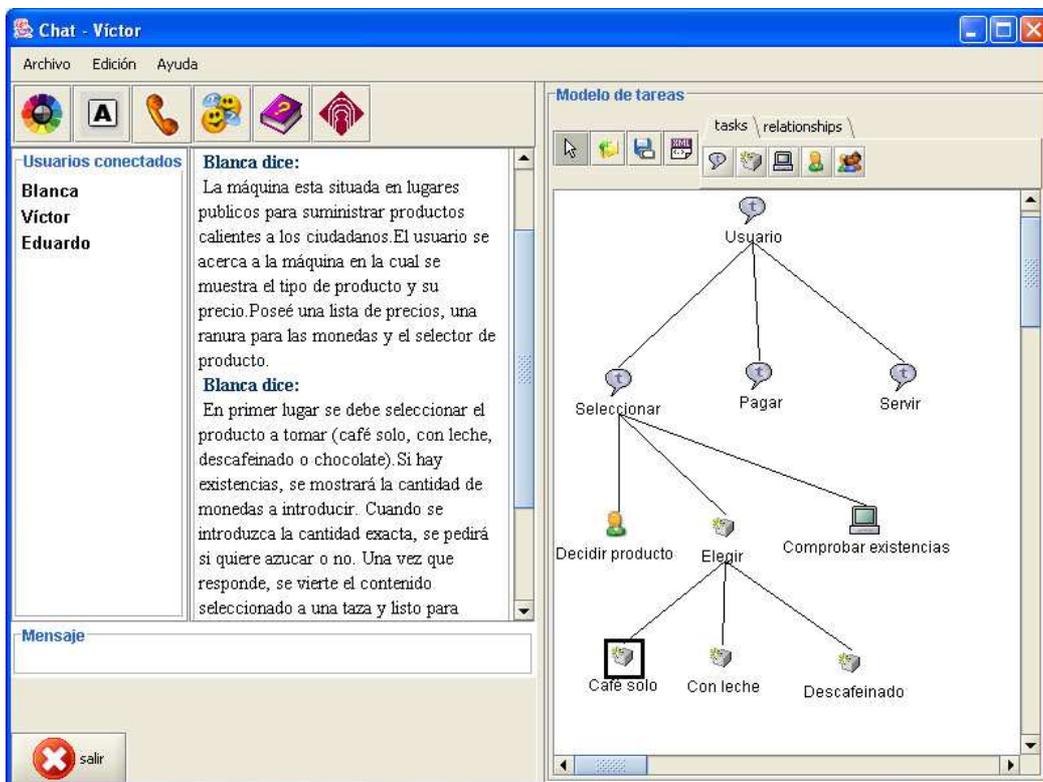


Figura 5-3. Resolución del problema con la herramienta CoCTT (Ejemplo 2)

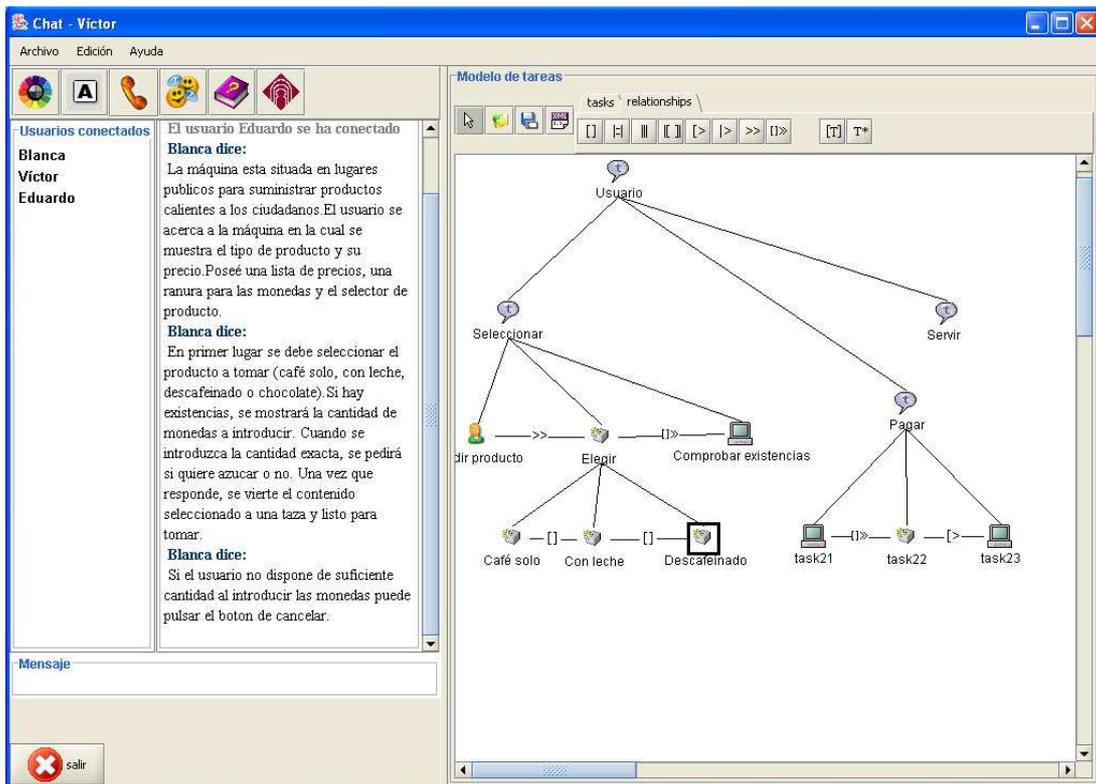


Figura 5-4. Resolución del problema con la herramienta coCTT (Ejemplo 3)

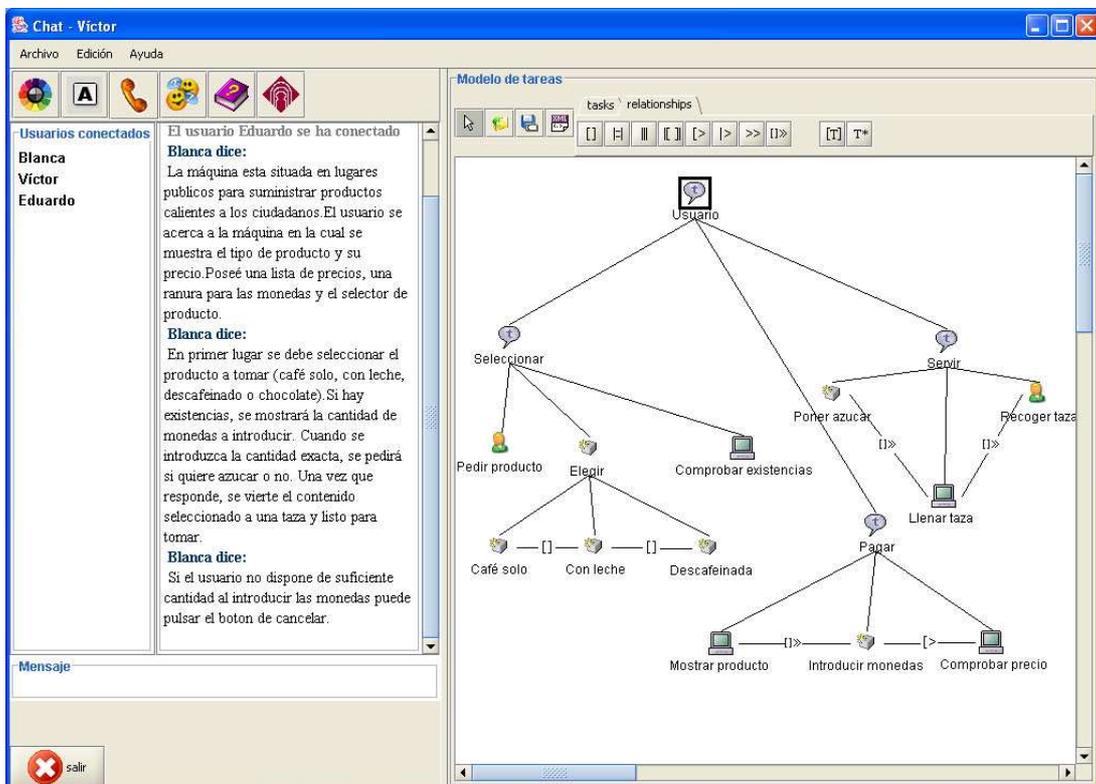


Figura 5-5. Resolución del problema con la herramienta coCTT (Ejemplo 4)

El usuario podría visualizar el código XML (Figura 5-6) que genera el diagrama dibujado, así como guardar (Figura 5-7) el diagrama para su posterior reutilización.

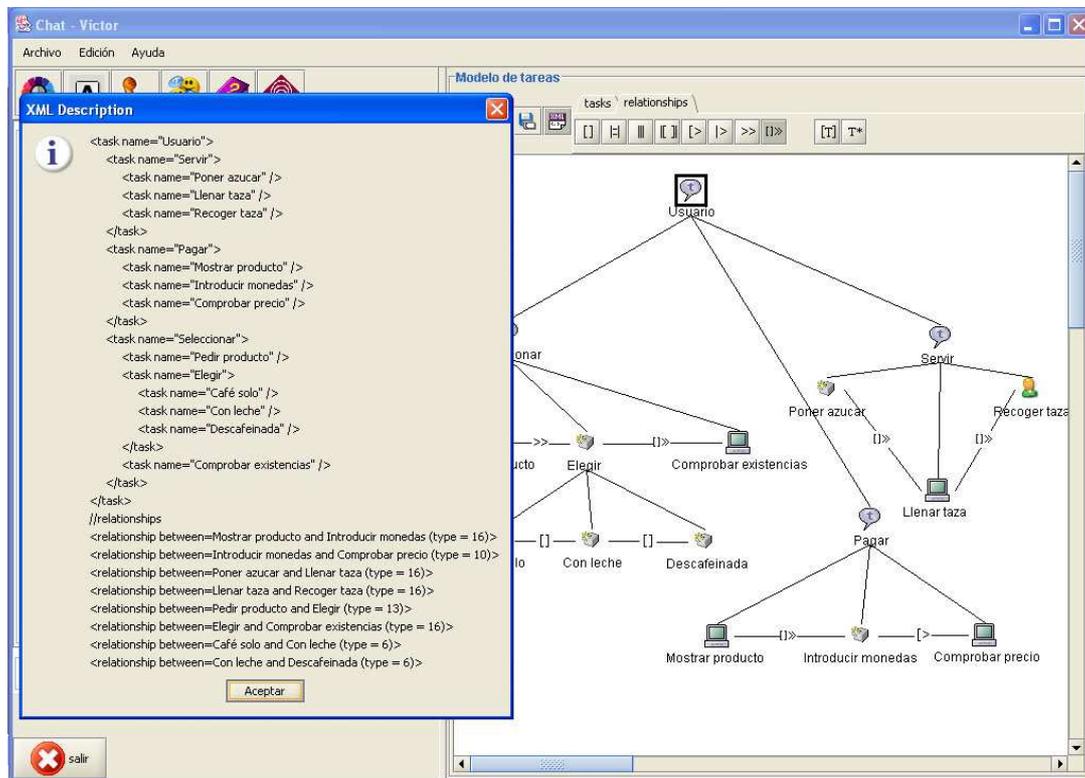


Figura 5-6. Consulta de la especificación XML asociada a un modelo de tareas

Tras la realización del ejemplo se llega a la conclusión de que para la elaboración de análisis de gran envergadura, los usuarios deben ponerse de acuerdo sobre quién será el responsable de "dibujar" la tarea en el panel en cada momento, pues sólo habrá una tarea seleccionada y solo se permitirá dibujar una tarea en un instante determinado. El problema puede aparecer si dos usuarios intentan dibujar tareas diferentes en ese mismo instante. La solución a este problema se propone como trabajo futuro en el Capítulo 6.

Por lo demás la herramienta permite una eficaz colaboración entre usuarios para realizar el análisis, disminuyendo notablemente el tiempo en la realización del modelo. Además, facilita la comprensión y al mismo tiempo que facilita la colaboración entre usuarios que se encuentran en distintas ubicaciones, podría ayudar también a los que estando en el mismo lugar de trabajo, necesitarán la participación o ayuda de otros para realizar el análisis de las tareas más dificultosas o menos comprensibles.

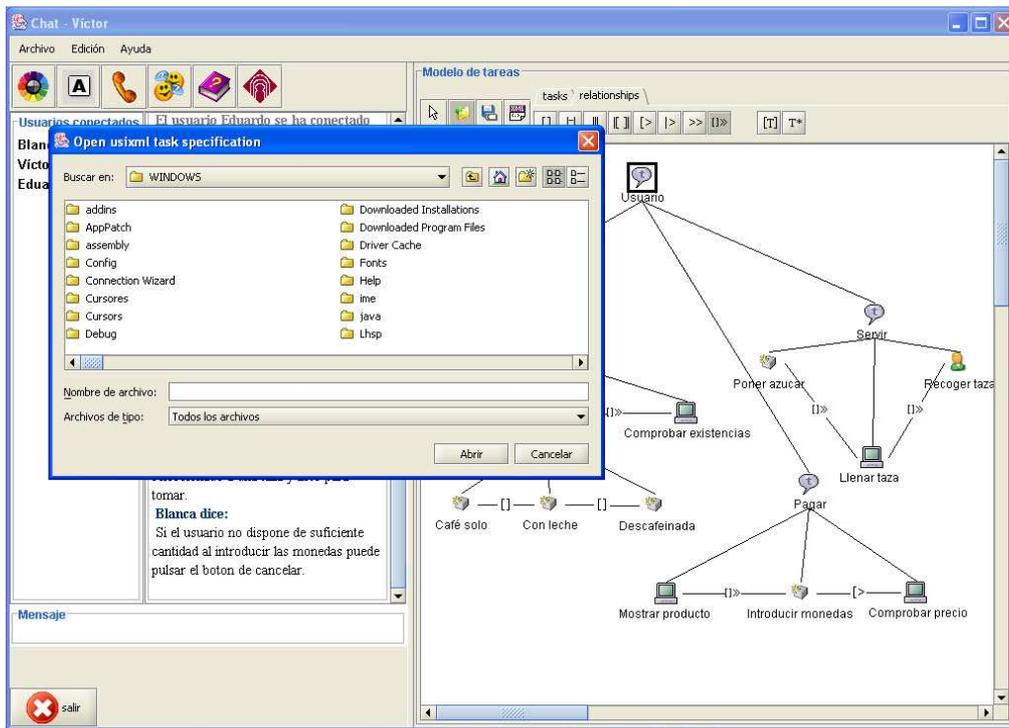


Figura 5-7. Invocación a la operación Guardar

5.2. CoCTT: EJEMPLO 2. PROPUESTA DE RESOLUCIÓN ANTE UN PROBLEMA AMBIGUO

Un problema en la construcción de modelos de tareas usando las relaciones de dependencia que vienen dadas por la notación CTT es la ambigüedad en las expresiones, como se puede observar en la Figura 5-8.

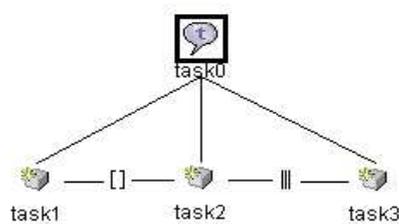


Figura 5-8. Situación de posible ambigüedad en la especificación de un modelo de tareas

La especificación $T1 \text{ [] } T2 \text{ ||| } T3$ puede ser interpretada de dos modos diferentes que deberían ser discutidos entre los desarrolladores para eliminar la ambigüedad

creada. La herramienta coCTT podría solucionar este problema permitiendo a los usuarios comunicarse de una forma fácil, precisa y descriptiva (Figura 5-9).

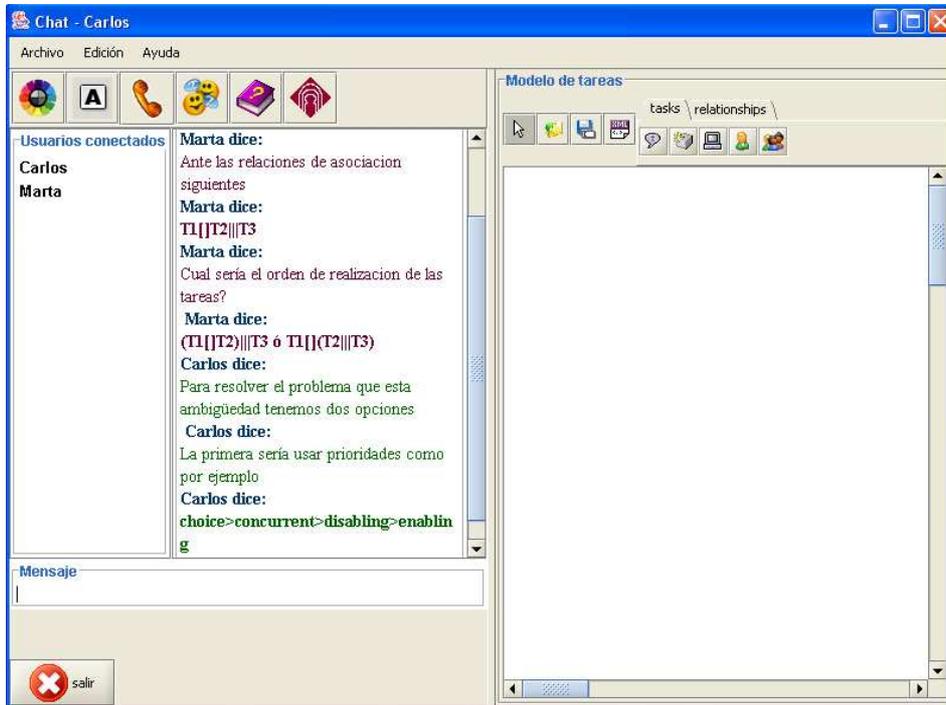


Figura 5-9. Caso práctico, análisis de tareas: Exposición de problema de ambigüedad

La ambigüedad podía resolverse de dos modos diferentes:

- i) Asignando prioridad a los operadores: $choice > concurrent > disabling > enabling$, la sucesión de tareas sería $(T1[]T2)|||T3$
- ii) Si el desarrollador no quiere introducir prioridad, otra posibilidad sería una nueva tarea abstracta, task D, la cual elimine la ambigüedad creada, (véase Figura 5-10). Esta solución puede ser adoptada sólo si la nueva tarea abstracta introducida representa una agrupación significativa de las tareas.

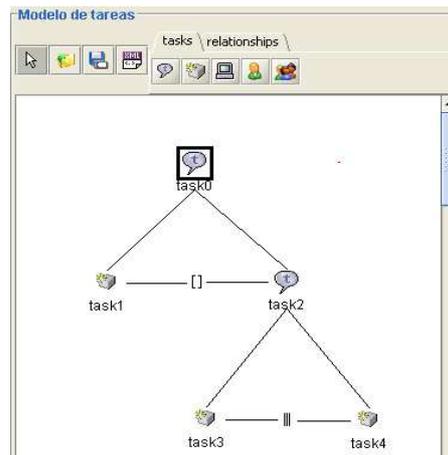


Figura 5-10. Caso práctico, análisis de tareas: resolución gráfica de problemas de ambigüedad

5.3. CoCTT: EJEMPLO 3. APRENDIZAJE ENTRE USUARIOS

Otra de las ventajas que supone la herramienta CoCTT es favorecer el aprendizaje entre usuarios. Los desarrolladores más experimentados en las tareas de modelado podrían ayudar a los usuarios recién incorporados en la realización de estas tareas sin necesidad de encontrarse en el mismo lugar siendo CoCTT un elemento de gran ayuda en el camino de aprendizaje común entre usuarios.

Las Figuras 5-11, 5-13, 5-14 y 5-15 y 5-15 ilustran un ejemplo práctico del aprendizaje llevado a cabo entre usuarios.

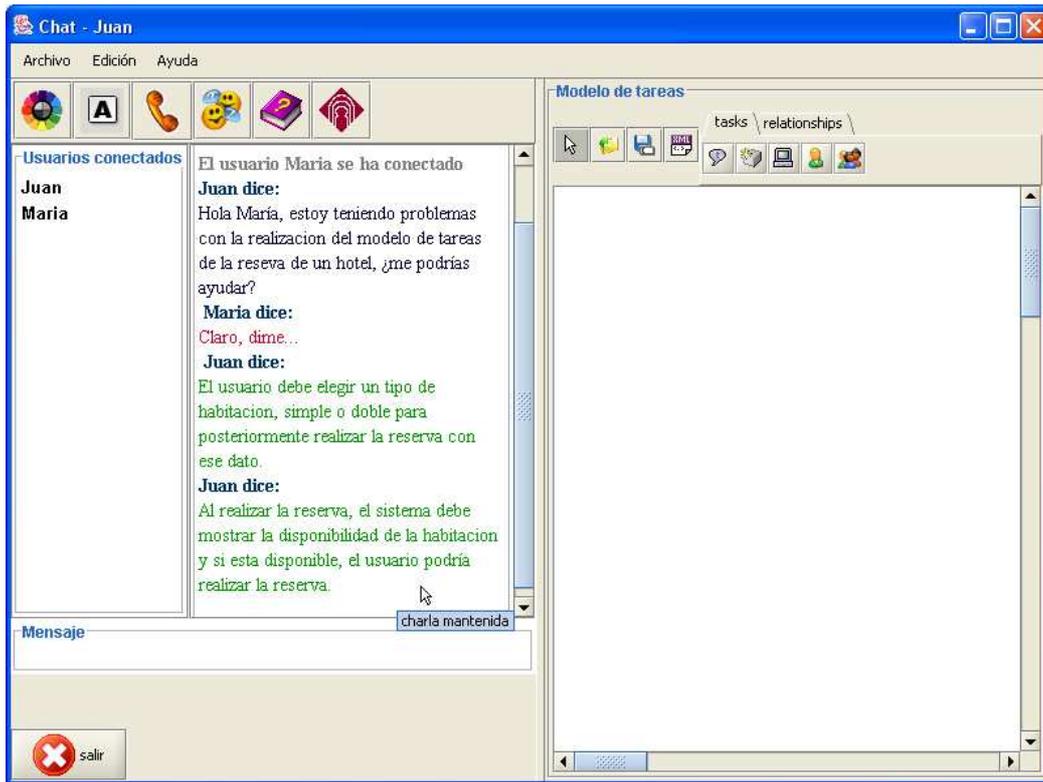


Figura 5-11. Caso práctico de análisis de tareas: exposición de problemas (1)



Figura 5-12. Caso práctico de análisis de tareas: resolución de problemas (2)

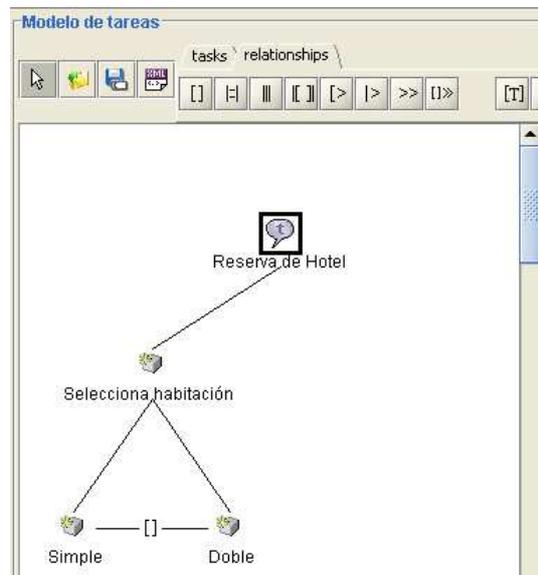


Figura 5-13. Caso práctico de análisis de tareas: resolución de problemas (3)

Chat - Maria

Archivo Edición Ayuda

Usuarios conectados
Juan
Maria

herencia y tareas de tipo interacción.
Juan dice:
 ¿Qué relación se establecería entre las tareas?
Maria dice:
De tipo elección
Maria dice:
 El usuario debe elegir entre habitacion simple o doble
Maria dice:
 La información del tipo de habitacion seleccionada habría que pasársela a la tarea Hacer reserva.
Maria dice:
Empiezo a dibujar el diagrama para que veas como sería
Maria dice:
 lo puedes terminar tu para ir aprendiendo

Mensaje charla mantenida

Modelo de tareas

```

    graph TD
      A[Reserva de un Hotel] --- B[Selección de Habitación]
      B --- C[Simple]
      B --- D[Doble]
  
```

Figura 5-14. Caso práctico de análisis de tareas: resolución de problemas (4)

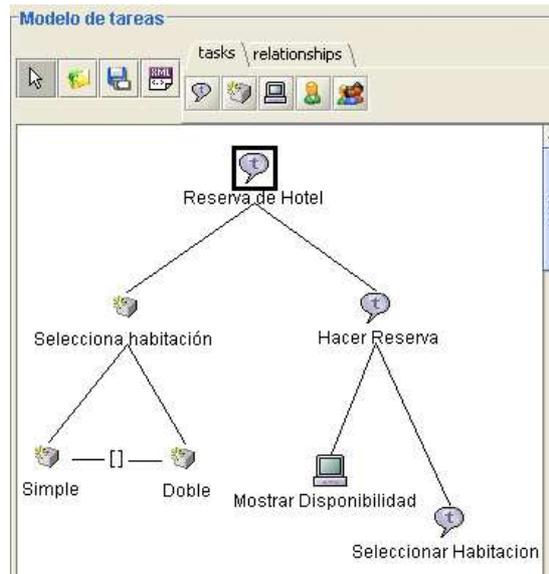


Figura 5-15. Caso práctico, análisis de tareas: resolución gráfica de problemas (5)

Como conclusión final tras haber realizado estos ejemplos, se puede decir que coCTT favorece las tareas de análisis en varios aspectos:

La ventaja principal es la ayuda que proporciona a los usuarios que tienen que trabajar conjuntamente eliminando la necesidad de que éstos deban estar reunidos para lograr un trabajo conjunto eficaz. Además de esta ventaja, también se pueden mencionar otras cuestiones favorables proporcionadas por la herramienta, como permitir la eliminación de ambigüedades en el momento exacto de la realización del análisis, fomentar el trabajo en equipo, la colaboración y cooperación entre usuarios de las que se hablaban en el capítulo 2, beneficiar el aprendizaje que pueden proporcionar los desarrolladores más experimentados al resto de usuarios...etc.

En el siguiente Capítulo se exponen las conclusiones finales sobre este Proyecto Final de Carrera y se habla de forma más detallada sobre las ventajas que esta herramienta aporta. Además, se hablará de las líneas de trabajo futuras que se pueden seguir para mejorarla.

Capítulo 6

Trabajo futuro y conclusiones

6.1. CONCLUSIONES

Como se ha comentado en capítulos anteriores, la aplicación desarrollada en este Proyecto permite la interacción y colaboración de un grupo de personas en la consecución de un trabajo común, sin necesidad de proximidad física, más concretamente en el modelado de tareas que se lleva a cabo durante la construcción de interfaces de usuario.

Este Proyecto ha sido una excelente oportunidad para consolidar conocimientos acerca del trabajo cooperativo mediante el Capítulo 2, introduciendo conceptos como CSCW y Groupware, así como presentando las mejoras e inconvenientes que esta forma de trabajo supone y las clasificaciones que se pueden encontrar hoy en día de las herramientas disponibles. También se han expuesto una serie de dinámicas para la realización del trabajo en grupo las cuales han permitido conocer la forma en la que trabajan hoy en día las organizaciones cuando es necesario un grupo numeroso de personas para la consecución de una labor común. Además, han sido expuestos términos claves en la realización de trabajos en grupo como colaboración, coordinación y cooperación, así como los factores que intervienen en este tipo de trabajo.

El Capítulo 2 también ha aportado información sobre cuestiones de análisis y modelado de tareas, elementos clave en la construcción de interfaces de usuario, permitiendo afianzar conocimientos sobre las distintas técnicas de modelado de tareas actuales haciendo una descripción más profunda sobre la notación CTT por ser la notación utilizada en este Proyecto Final de Carrera.

El Capítulo 3 ha servido para dar a conocer algunas herramientas características en el modelado de tareas y presentar la nueva herramienta colaborativa que se ha desarrollado: coCTT.

El Capítulo 4 ha sido útil para presentar la aplicación realizada, coCTT, utilizando los conocimientos adquiridos en el Capítulo 2. Se ha presentado el diseño de la aplicación permitiendo afianzar conocimientos relativos a la Ingeniería del

Software. coCTT permite la colaboración entre usuarios para realizar especificación y el modelado de tareas. CoCTT espera ser una herramienta que al mismo tiempo que fomenta el trabajo en grupo, facilite el aprendizaje entre usuarios, así como la rápida comunicación y cooperación entre estos. En este capítulo se hace referencia al manual de usuario que pretende facilitar el manejo de la herramienta.

En el Capítulo 5 se incorporan algunos ejemplos que han permitido dar una visión más detallada sobre la realización de modelos de tareas así como del uso de la herramienta, permitiendo realizar un estudio sobre las ventajas que esta aporta así como sus limitaciones, las cuales han servido para reflexionar a cerca de las posibles mejoras que podrían llevarse a cabo sobre la herramienta en un futuro.

A modo de conclusión personal se puede decir que este Proyecto me ha ayudado a afianzar y consolidar conocimientos en análisis y programación orientada a objetos, así como en interfaces de usuario y trabajo cooperativo.

6.2. TRABAJO FUTURO

En el Capítulo 4 se comentó la posibilidad de realizar el Proyecto utilizando un modelo basado en objetos en vez de el finalmente utilizado, cliente-servidor. Como alternativa a los sockets se podrían haber utilizado sistemas este tipo de sistemas como son RMI y CORBA.

RMI (Invocación Remota de un Método) fue el primer Framework con la idea de crear sistemas distribuidos que apareció en Java. Además, viene integrado en cualquier máquina virtual Java posterior a la versión 1.0 y está pensado para hacer fácil la creación de sistemas distribuidos a partir de una aplicación cuyas clases están ya implementadas. La invocación de métodos remotos permite que un objeto que se ejecuta en una máquina puede invocar métodos de un objeto que se encuentra en ejecución bajo el control de otra máquina (por supuesto, no hay problemas para las relaciones entre objetos cuando ambos son locales). En definitiva, RMI permite crear e instanciar objetos en máquinas locales y al mismo tiempo crearlos en otras máquinas (máquinas remotas), de forma que la comunicación se produce como si todo estuviese en local. RMI se puede convertir

así en una alternativa muy viable a los sockets de bajo nivel con una serie de particularidades destacables:

- RMI permite abstraer las interfaces de comunicación llamadas locales, no necesitamos fijarnos en el protocolo y las aplicaciones distribuidas son de fácil desarrollo.
- RMI permite trabajar olvidándose del protocolo.

Los sistemas remotos requieren que las partes que lo componen y que se ejecutan en diferentes espacios de direcciones (posiblemente en diferentes máquinas), tengan la capacidad de comunicarse entre sí. Una solución a esta problemática (y la utilizada en este Proyecto) son los sockets, ya que tienen la capacidad de comunicar dos procesos, ya sea mediante datagramas o flujos de datos (streams). Sin embargo, los sockets requieren que las aplicaciones implanten sus propios protocolos para codificar y decodificar los mensajes que intercambian, lo que introduce una problemática diferente a la naturaleza del problema a resolver y aumenta la posibilidad de errores durante la ejecución.

El empleo de objetos distribuidos implica varias ventajas como la orientación a objetos misma, movilidad de las aplicaciones Java, los patrones de diseño, la seguridad, etc.

Por el contrario, los objetos distribuidos también tienen desventajas con respecto a los procedimientos remotos y sockets siendo la principal el rendimiento. Esta desventaja es análoga a la del mismo lenguaje Java, con respecto a los lenguajes completamente compilados como C, pero de la misma manera que en este caso, todas las características positivas del modelo de objetos distribuidos de Java, lo pueden hacer una alternativa bastante interesante con respecto a sus contrapartes procedimentales de más bajo nivel.

Una herramienta que podría haber sido útil en la realización de este Proyecto es la herramienta JGroup (hasta hace poco conocida como JavaGroups); es una herramienta que soporta comunicación con grupos cerrados y está escrita enteramente en Java con la finalidad de explotar, en lo posible, todos los beneficios de este lenguaje. JGroups proporciona soporte para fiabilidad, atomicidad y causalidad. Puede ser considerado como una API en su totalidad, lo que le atribuye

una enorme facilidad para su expansión e integración con otras herramientas para comunicación en grupo.

Para unirse a un grupo y enviar mensajes, un proceso tiene que crear primero un canal y conectarse a él especificando, para ello, el nombre del grupo (todos los canales con el mismo nombre forman un grupo). Las propiedades del canal son fijadas en el momento de su creación, lo cual origina también la creación de la pila de protocolos sobre la que se ubica el canal.

La característica que otorga más potencia a JGroups es, precisamente, la flexibilidad de su pila de protocolos: mediante la elección, mezcla y sustitución de los protocolos, diferentes requerimientos de aplicación y características de red pueden ser satisfechos. JGroups dispone de un variado número de protocolos (transporte, fragmentación, envío y radiado fiable, detección de fallos, orden, membresía, etc.), aunque cualquier otro puede ser específicamente implementado.

JGroups dispone de dos interfaces para los usuarios: una de bajo nivel (canal) que ofrece la funcionalidad básica para comunicación en grupo y; otra, de alto nivel (formada por bloques construidos a partir de la combinación de elementos más básicos), que proporcionan al programador un nivel de abstracción mayor, facilitando la tarea de desarrollar aplicaciones.

Otra alternativa que se podría incorporar a este proyecto es el uso de una pizarra compartida en lugar del editor de tareas. Una pizarra permite a un grupo de usuarios compartir una superficie de dibujo, que podría apoyar al sistema de comunicación síncrona construido.

Por último, otra posible mejora que se podría proponer es la implementación mecanismos de sincronización (reloj/tiempo, paso de testigo...) para que cada usuario tuviera un tiempo mínimo en la realización del modelado de tareas, de modo que no se pudieran entorpecer en el momento de que cada uno de ellos, individualmente, realizase esta labor.

Estas ideas podrían servir para enriquecer y fortalecer la herramienta construida en un futuro.

Bibliografía

REFERENCIAS BIBLIOGRÁFICAS

[Annett, 1967] Annett M. The binomial distribution of right, mixed and left handedness.

[Ayala, 1997] Ayala, G. "Intelligent Agents for Supporting the Effective Collaboration in CSCL Environment". Tesis doctoral, Universidad de Tokushima, Japón, 1996.

[Baec,1993]. Baecker, R. M. "Readings in Groupware and Computer-Supported Cooperative Work". Morgan Kaufmann Publishers, 1993.

[Bannon et.al., 1989]. Bannon, L. & Schmidt, K. (1989) CSCW: Four Characters in Search of a Context. In Proc. First European Conf. on CSCW, Gatwick, UK, Sept. 1989. (Reprinted in J. Bowers & S. Benford (Eds.) Studies in Computer Supported Cooperative Work: Theory, Practice and Design. pp. 3-16. (Amsterdam: North-Holland).

[Card, Moran et al. 1983] Goals, Operators, Methods, Selection Rules.

[Coleman,1995] COLEMAN, David, KHANNA, Raman (eds). Groupware: Technology and Applications. Prentice Hall, Upper Saddle River, NJ, 1995.

[CSCW, 1984] Irene Grief and Paul Cashman, CSCW Workshop, 1984.

[Dave Chaffney] Groupware, Workflow and Intranets. Reengineering Software. Chaffey Dave, Ed. Digital Press, 1998.

[ELLIS 91] Ellis C. "Groupware: Some Issues and Experiences" en Communications of the ACM.

[Ellis et.al., 1993] ELLIS, C., GIBBS, S. y REIN, G. Groupware: Some Issues and Experiences. En: Readings in Groupware and Computer-Supported Cooperative Work. BAECKER, R., San Mateo CA: Morgan Kauffmann, 1993: pp. 9-28.

Francisco Montero, Víctor López-Jaquero, María Lozano, Pascual González, IDEALXML: un entorno para la gestión de experiencia relacionada con el desarrollo hipermedial, Universidad de Castilla la Mancha.

G. Gerónimo, V. Canseco, Breve Introducción a los Sistemas Colaborativos: Groupware & Workflow Universidad Tecnológica de Mixteca, México.

[Gol 94] GOLDBERG A. P. Groupware Lectura Notes.

[Grudin 1994a] GRUDIN, Jonathan. CSCW: History and Focus. IEEE Computer, 27 (5). Mayo, 1994: pp. 19-26.

[Grudin J, 1991] Groupware and Cooperative Work: Problems and Prospects. The Art of Human-Computer Interface Design,

[Grudin 1994b] Grudin, Jonathan. Groupware and Social Dynamics: Eight Challenges for Developers. Communications of the ACM, 1 (37). Enero 1994: pp. 92-105.

Harstson, R. and P. Gray (1992). "Temporal Aspects of Tasks in the User Action Notacion." Human Computer Interaction 7: 1-45.

Hix and Harstson 1994 Hix, D. and R. Harstson (1994). Ideal: An environment to support usability engineering. 1994 East-West International Conference on HCI, St. Petersburg (Russia).

Hudson, John et al. 2000 Hudson, S., B. John, et al. (2000). A Tool for Creating Predictive Performance Models from User Interface Demonstrations. UIST'2000, ACM Press.

[CSCW, 1984] Irene Grief and Paul Cashman, CSCW Workshop, 1984.

Jacobson, I, Booch, G., Rumbaugh, J., El Proceso Unificado de Desarrollo de Software, Addison-Wesley, 2000.

[Johansen, 1988] JOHANSEN, R. Groupware: Computer Support for Business Teams. New York: The Free Press, 1988.

[Jonson-Lenz, 1981], Johnson-Lenz, Peter and Trudy Johnson-Lenz. 1981. Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium. In Hiltz, S. And Kerr, E. (eds.). Studies of Computer-Mediated Communications Systems: A Synthesis of the Findings, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, New Jersey.

Manuel Ortega, José Bravo, Trabajo Cooperativo con Ordenador, Universidad de Castilla la Mancha.

[Malone 87]. "Computer Support for Organizations: Towards an organizational science". Thomas Malone. MIT. 1987.

Miguel A. Redondo, Grupo CHICO Modelos basados en tareas. Lenguajes para el modelado.

[Montero, 2005] Tesis doctoral Integración de calidad y experiencia en el desarrollo de interfaces de usuario dirigido por modelos, Universidad de Castilla la Mancha.

[Ortega, 2001]ORTEGA, Manuel, BRAVO, José. Trabajo Cooperativo con Ordenador. En: La interacción persona-ordenador (Libro electrónico de la Asociación Interacción Persona Ordenador, AIPO), Lorés, Jesús (ed), Lleida, 2001.

[Paternò, 1999] Paternò, F. Model-based Design and Evaluation of Interactive Applications. Springer Ver-lag, November 1999, ISBN 1-85233-155-0.

Paternò, F., Mancini, C., Meniconi, S.: ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. Proceeding of Interact '97. (July 1997) 362-369.

[SAA, 1997] Saadoun M. El proyecto groupware. De las técnicas de dirección a la elección de la aplicación groupware. Ediciones Gestión 2000, S.A. Barcelona, 1997.

Victor M. R. Penichet, Maria D. Lozano, Jose A. Gallud, Hacia una Propuesta Metodológica para Entornos Colaborativos.

[Wilson, P. 91] Computer Supported Cooperative Work. Kluwer Academic Publishers, Great Britain.

[Wilson, 1991] Wilson, P. (1991). Computer supported cooperative work : an introduction. Oxford, England Norwell, MA, Intellect ; Sold and distributed in the U.S.A. and Canada by Kluwer Academic Publishers.

Apéndice A

Manual de usuario de coCTT

A.1. Pantalla de inicio

Para iniciar la aplicación se debe de conectar al servidor que se encontrará en una máquina determinada. El resto, serán maquinas clientes que se conectarán entre ellas a través de esta. Una vez iniciado el servidor aparecerá un mensaje como el que se muestra en la Figura A-1.

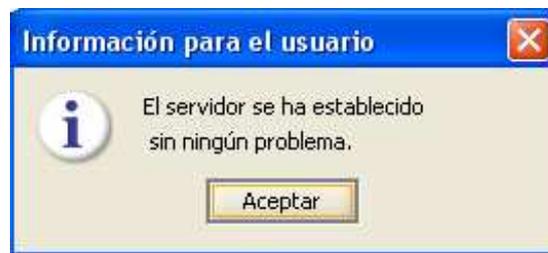


Figura A-1. Establecimiento del servidor CoCTT

Una vez iniciado el servidor, nos dispondremos a ejecutar la aplicación en cualquier máquina cliente. Se mostrará una ventana de inicio tal y como se muestra en la Figura A-2.



Figura A-2. Inicio de sesión

El nickname o nombre de usuario no debe haber sido escogido por otro compañero anteriormente. Si este caso ocurre, la herramienta advierte al usuario de este hecho indicándole que introduzca otro nombre, (véase Figura A-3).



Figura A-3. Nombre de usuario ya está utilizado

Una vez introducido el nickname, la máquina cliente se conecta con el servidor comprobando si el nombre es válido, si es así, el usuario accede a la aplicación, la cual se muestra en la Figura A-4.

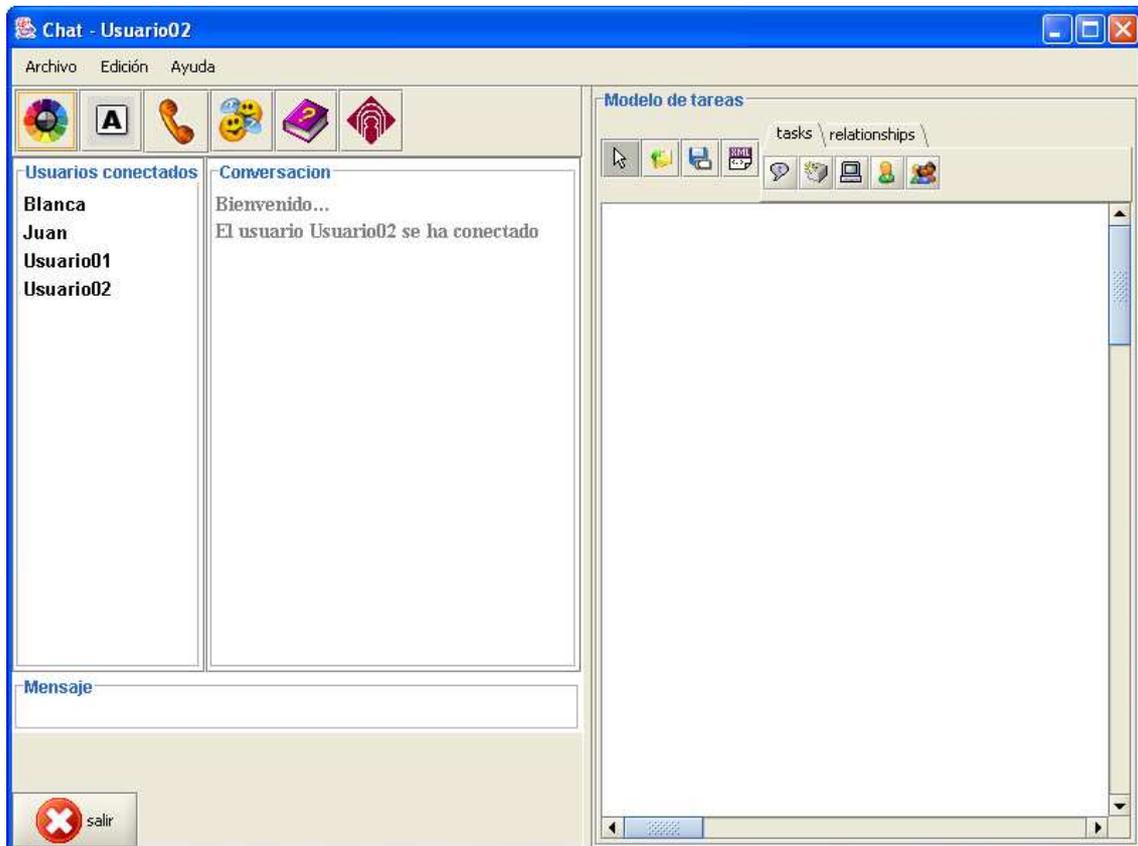


Figura A-4. Vista al acceder a la aplicación

La aplicación se compone de dos partes principales: la *conversación* y la parte dedicada a la construcción del *modelo de tareas*. En los apartados siguientes se describirá el funcionamiento de ambas.

A.2. Conversación

La conversación se compone de tres áreas principales, como se pueden ver en la Figura A-5:

- Menú: Situado en la parte superior.
- Usuarios conectados: Indican todos los usuarios conectados en el sistema.
- Chat: Editor por el que los usuarios pueden mantener una conversación.



Figura A-5. Partes en las que se divide el editor de Chat en la aplicación CoCTT

A.2.1. Menú asociado a los botones

En la parte superior se encuentran situados los **botones**, Figura A-6, con las siguientes funcionalidades:



Figura A-6. Botones del editor de Chat

- **Color del texto:** El usuario puede cambiar el color del texto a su gusto para poder diferenciar su diálogo del diálogo de los demás participantes. Para ello se le muestra una paleta de colores donde podrá escoger el color deseado. En la Figura A-7 se puede observar este comportamiento.

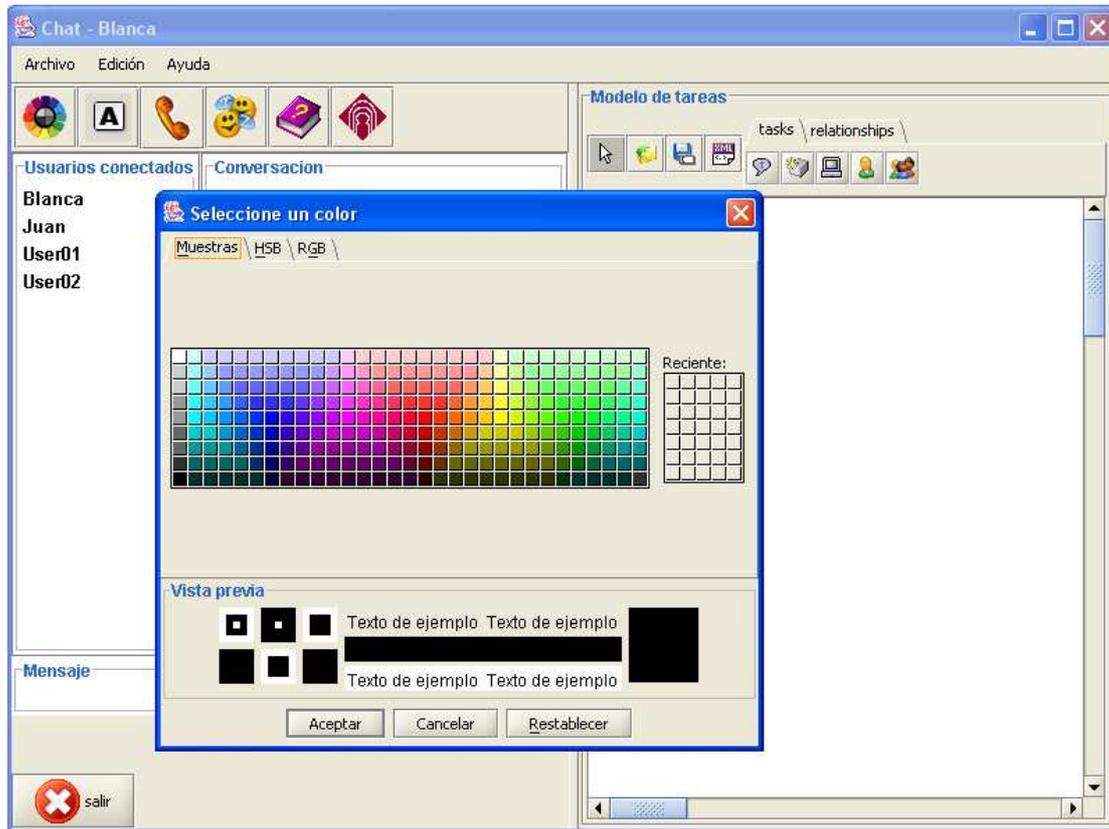


Figura A-7. Elección de color del texto de comunicación entre usuarios

- **Negrita:** Destaca el texto enviado por el usuario. Con ella es posible diferenciar párrafos que puedan tener más relevancia sobre los demás haciendo posible que el resto de compañeros les presten mayor atención. Para obtener el texto en negrita simplemente hay que seleccionar el botón correspondiente en el cuadro de diálogo, como muestra la Figura A-8.

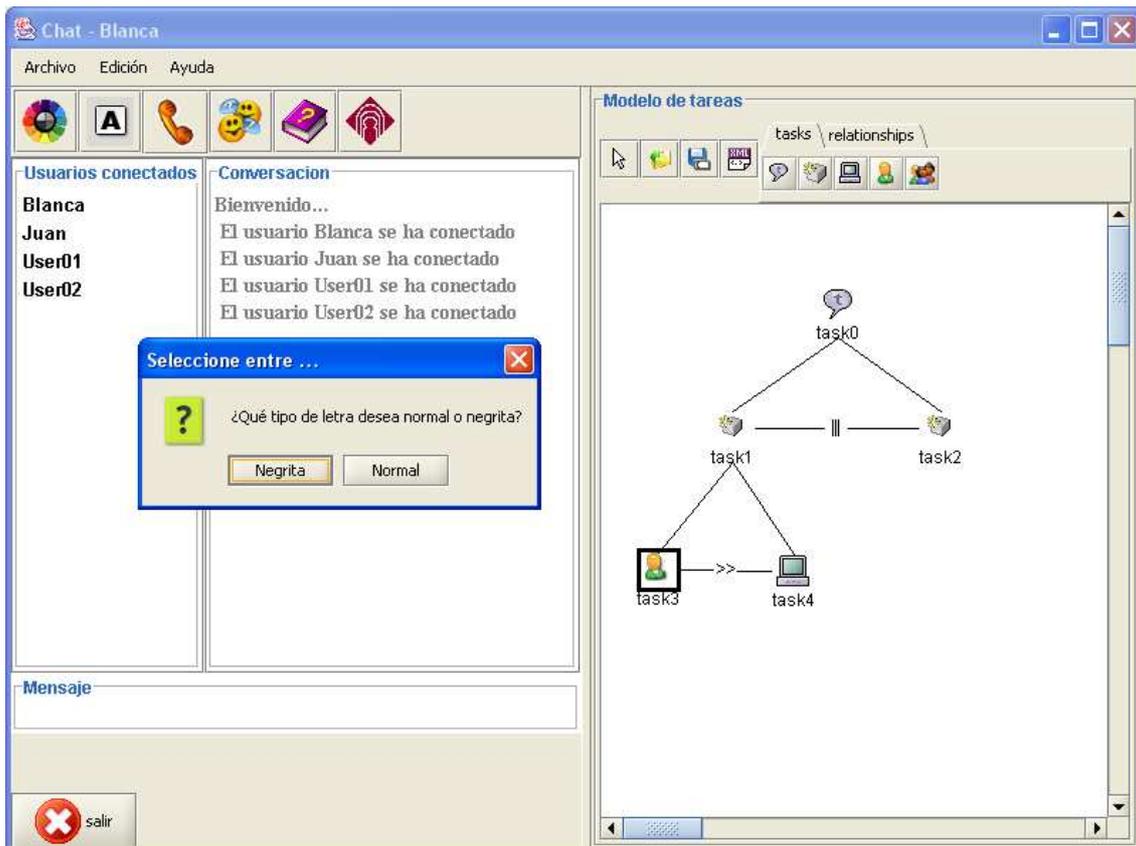


Figura A-8. Elección del texto en negrita

- **Iconos:** El sistema dispone de dos iconos que serán útiles para el usuario cuando este quiera comunicar alguna acción. *Icono Teléfono* que muestra al usuario en estado de "hablando por teléfono" y el icono *charla privada* que advierte al resto de usuarios de que se está manteniendo una conversación privada.
- **Ayuda:** Muestra un manual de ayuda de la aplicación.
- **Acerca de:** Muestra las referencias del autor de la aplicación (véase Figura A-9).

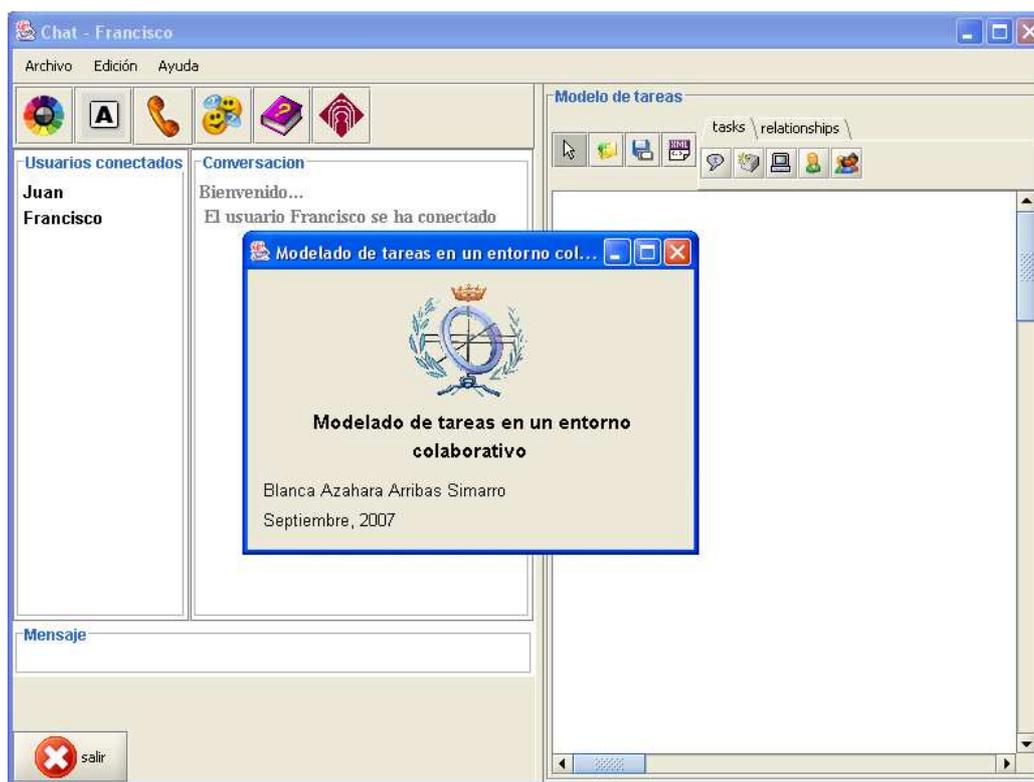


Figura A-9. Funcionalidad del botón Acerca de...

A.2.2. Usuarios conectados

Cada vez que un nuevo usuario se conecte en el sistema esta acción quedará reflejada en el panel así como en el caso que el usuario salga del sistema, (Figura A-10). De este modo, los usuarios que estén trabajando sabrán en todo momento qué compañeros están on-line.

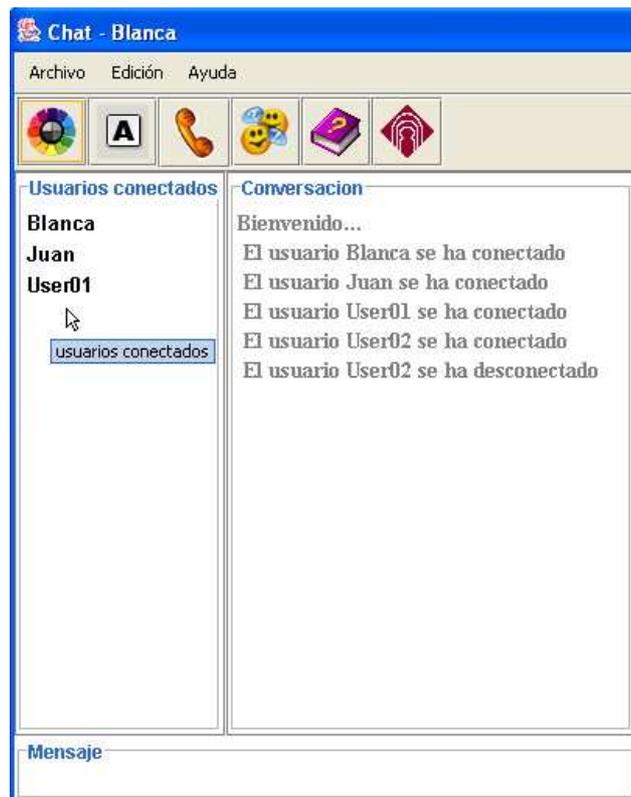


Figura A-10. Vista de la desconexión del sistema de un usuario

A.2.3. Editor de Chat

El Chat (Figura A-11), sirve a los usuarios para comunicarse entre ellos, resolver dudas, cuestiones, posibles mejoras, solucionar problemas...etc. El editor que aparece por defecto es un editor general al que tendrán acceso todos los usuarios conectados.

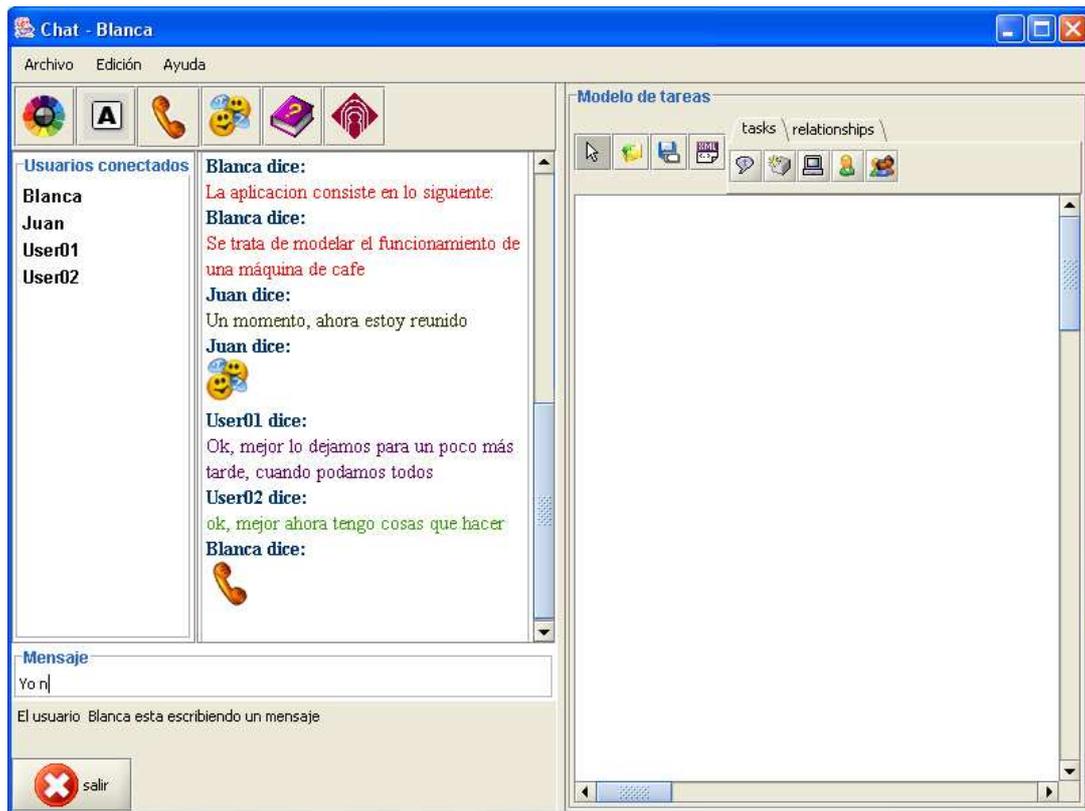


Figura A-11. Facilidad de chat proporcionada

Como se observa en la Figura A-11, siempre que un usuario este escribiendo un mensaje, este hecho es indicado en la parte posterior del área destinada a escribirlo.

Otra utilidad del Chat es la **conversación privada**: Es útil en el caso de que un usuario quiera comentar hechos puntuales sobre la realización del modelo de tareas sin tener que entorpecer la conversación llevada a cabo por el resto. Sólo se podrá mantener una única conversación privada a la vez pues el objetivo de esta herramienta es la construcción de un modelo de forma colaborativa y la inclusión de múltiples conversaciones privadas podría entorpecer este hecho desviando la atención de los usuarios.

Esto no quiere decir que un usuario no se pueda comunicar con el resto de forma privada. El funcionamiento es el siguiente:

Una conversación privada se establece haciendo clic sobre el nombre del usuario deseado, en la parte correspondiente al panel de usuarios conectados. Se abrirá entonces una ventana tal y como muestra la Figura A-12 que permitirá al emisor comunicarse de forma privada con el receptor del mensaje:



Figura A-12. Ventana para mantener un Chat privado

A través de ella el usuario se podrá comunicar de forma privada con su receptor. Hay que tener en cuenta que una vez que se haya establecido una conversación privada, el usuario ya no podrá hacer clic sobre el nombre de otro usuario distinto, pues el sistema está configurado para mantener una y solo una conversación privada. En el caso de que desee hablar de este modo con otro usuario deberá cerrar la venta que corresponde a la conversación privada y de este modo podrá volver a hacer clic sobre el nuevo receptor con el que quiere conversar. No obstante, el sistema avisa al usuario cuando otro desee hablar con él de este modo.

Para indicar que el receptor está ocupado manteniendo una conversación privada con otra persona, aparecerá un mensaje de texto indicando este hecho, tal y como se muestra en la Figura A-13. Al receptor también se le indica que un usuario quiere mantener una conversación privada con él, (Figura A-14).

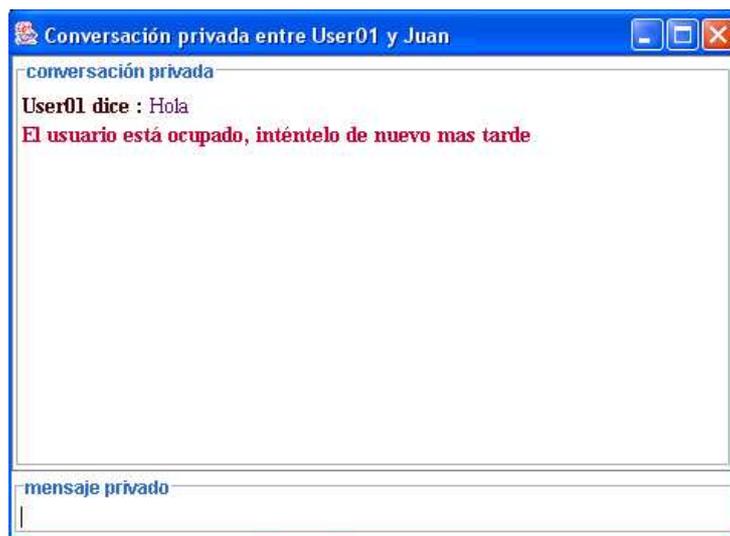


Figura A-13. Advertencia de usuario ocupado en Chat privado



Figura A-14. Advertencia de petición de conversación privada

A.3. Editor de tareas de modelado

La parte correspondiente al editor se presenta se muestra en la Figura A-15 y consta de las siguientes partes:

- Gestión del modelo
- Creación de tareas
- Creación de relaciones
- Propiedades de tareas y relaciones

- Movimiento
- Borrado
- Actualización inmediata

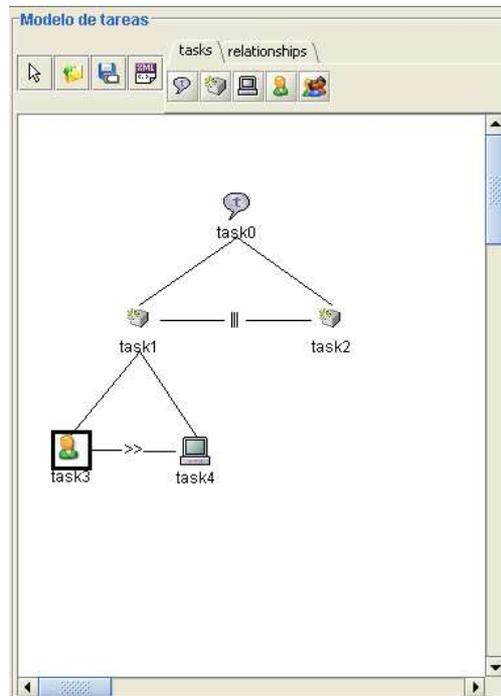


Figura A-15. Editor CTT colaborativo

A.3.1. Editor de Chat

Para gestionar el diagrama de tareas creado se dispone de botones, Figura A-16 y Tabla A-1.



Figura A-16. Botones para la gestión del modelo de tareas

Gestión del modelo	
	Seleccionar y movimiento de tareas y relaciones
	Abrir ficheros. Una vez abierto un fichero, este se abrirá en el resto de sesiones
	Guardar ficheros. Se guardan en formato xml
	Ver el código xml del diagrama de modelado

Tabla A-1. Botones para la gestión de modelos de tareas

A.3.2. Crear tareas

En la parte superior se encuentran los botones que permiten realizar las tareas de modelado, Figura A-17. Cada botón tiene una funcionalidad determinada mostrada en la Tabla A-12.



Figura A-17. Paleta de tareas

Task	
	Tarea abstracta: Requieren acciones complejas y por ello no es fácil decidir donde se van a realizar exactamente. Son tareas que serán decompuestas en un conjunto de nuevas sub-tareas.
	Tareas de interacción: Son tareas que realiza el usuario interactuando con la aplicación.
	Tareas de aplicación: Tareas realizadas por la aplicación. Pueden obtener información interna del sistema o producir información hacia el usuario.
	Tareas de usuario: Tareas realizadas completamente por el usuario, son tareas cognitivas o físicas que no interactúan con el sistema.
	Tareas cooperativas: Tareas que requieren la participación y cooperación de distintos usuarios para realizar actividades

Tabla A-2. Tipos de tareas

A.3.3. Crear relaciones

El sistema dispone de otra paleta correspondiente a las relaciones que pueden establecerse entre las tareas, como se muestra en la Figura A-18. En la Tabla A-3 se muestran los tipos de relaciones que pueden establecerse entre las tareas así

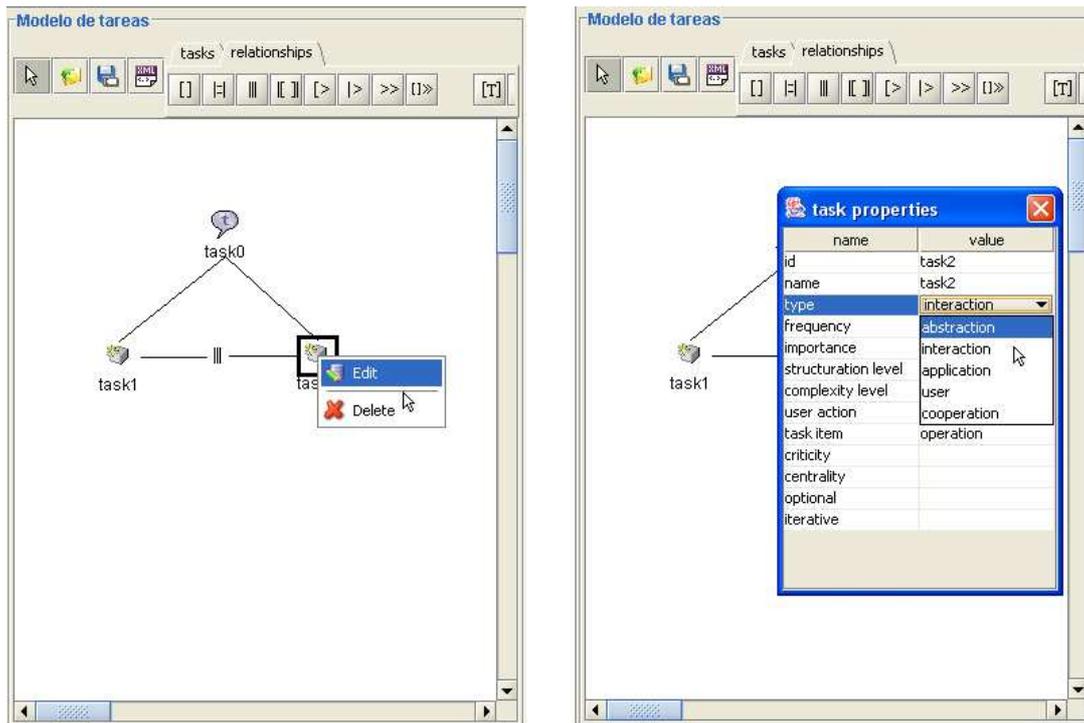


Figura A-19. Propiedades asociadas a las tareas

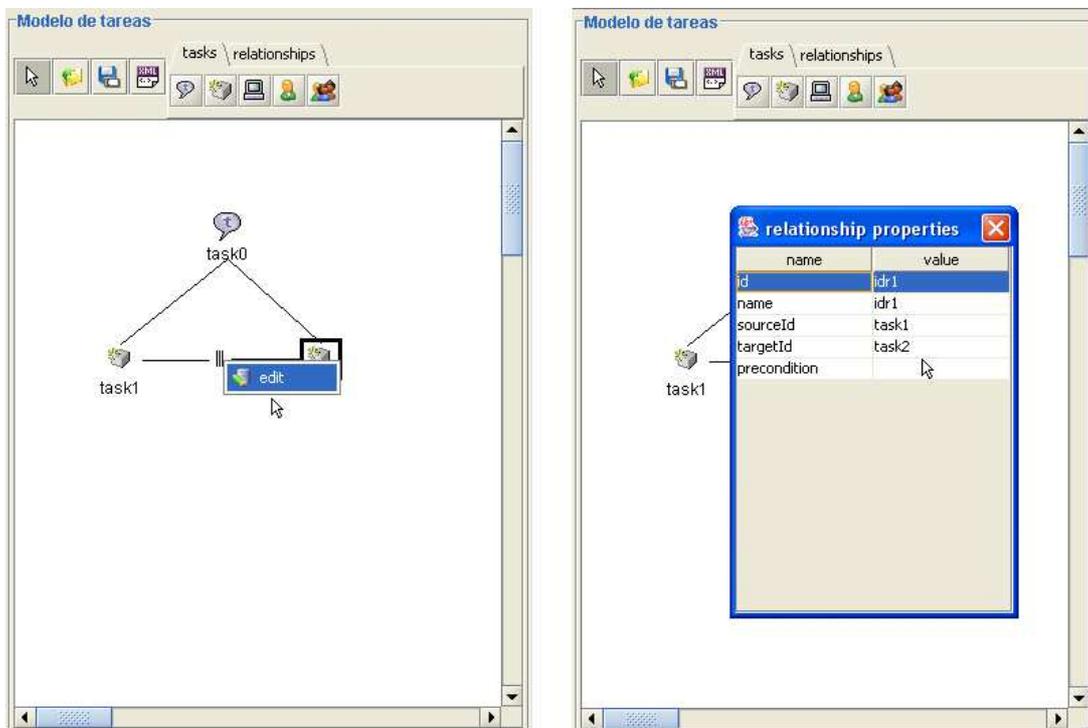


Figura A-20. Propiedades de las relaciones temporales entre tareas

A.3.5. Movimiento

Otra acción que puede realizarse en el editor de tareas es el movimiento de estas. Para ello tras pulsar el botón de selección se seleccionará la tarea a mover y se desplazará esta hasta el lugar deseado. Para dejar fija la tarea se pulsará el botón derecho del ratón. El cambio será actualizado en el resto de clientes.

A.3.6. Borrado

Las tareas pueden ser borradas. Para ello habrá se selecciona la tarea a borrar, botón derecho y Delete tal y como muestra la Figura A-21.

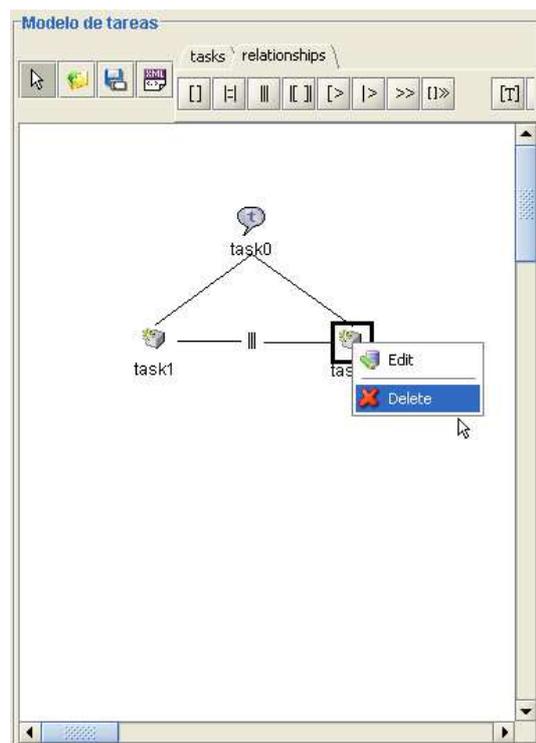


Figura A-21. Borrado de tareas

Hay que tener en cuenta que al borrar una tarea se borrará también las relaciones asociadas a esas tareas, así como sus tareas hijas en el caso de que esta las tuviera y el cambio se verá reflejado en el resto de clientes.

A.3.6. Actualización inmediata

Cualquier modificación realizada sobre el panel de modelado ya sea crear tareas, relaciones, modificar propiedades, abrir ficheros, mover o borrar será automáticamente actualizada en el resto de aplicaciones clientes.